

Multi-View Diffusion Maps

Ofir Lindenbaum¹ Arie Yeredor¹ Moshe Salhov² Amir Averbuch²

¹School of Electrical Engineering, Tel Aviv University, Israel

²School of Computer Science, Tel Aviv University, Israel

Abstract

In this paper, a reduced dimensionality representation is learned from multiple views of the processed data. These multiple views can be obtained, for example, when the same underlying process is observed using several different modalities, or measured with different instrumentation. The goal is to effectively utilize the availability of such multiple views for various purposes such as non-linear embedding, manifold learning, spectral clustering, anomaly detection and non-linear system identification. The proposed method, which is called multi-view, exploits the intrinsic relation within each view as well as the mutual relations between views. This is achieved by defining a cross-view model in which an implied random walk process is restrained to hop between objects in the different views. This multi-view method is robust to scaling and it is insensitive to small structural changes in the data. Within this framework, new diffusion distances are defined to analyze the spectra of the implied kernels. The applicability of the multi-view approach is demonstrated for clustering, classification and manifold learning using both artificial and real data.

I. INTRODUCTION

High dimension big data exist in various fields and it is difficult to analyze them as is. Extracted features are useful in analyzing these datasets. Some prior knowledge or modeling is required in order to identify the essential features. On the other hand, dimensionality reduction methods are purely unsupervised aiming to find a low dimensional representation that is based on the *intrinsic geometry* of the analyzed dataset that includes the connectivities among multidimensional data points within the dataset. A “good” dimensionality reduction methodology reduces the complexity of a data processing while preserving the coherency of the original data such that clustering, classification, manifold learning and many other data analysis tasks can be applied effectively in the reduced space. Many methods such as Principal Component Analysis (PCA) [1], Multidimensional Scaling (MDS) [2], Local Linear Embedding [3], Laplacian Eigenmaps [4], Diffusion Maps (DM) [5] and more have been proposed to achieve dimensionality reduction that preserve its data coherency. Exploiting the low dimensional representation yields various applications such as face recognition that is based on Laplacian Eigenmaps [6], Non-linear independent component analysis with DM [7], Musical Key extraction using DM [8], and many more. The DM framework extends and enhances ideas from other methods by utilizing a stochastic Markov matrix that is based on local affinities between multidimensional data points to identify a lower dimension representation for the data. All the mentioned methods do not consider the possibility of having more than one view to represent the same process. An additional view can provide meaningful insight regarding the dynamical process that has generated and governed the data.

In this paper, we consider learning from data that is analyzed by multiple views. The goal is to effectively utilize multiple views such as non-linear embedding, multi-view manifold learning, spectral clustering, anomaly detection and non-linear system identification to achieve better analysis of high dimension big data. Most dimensionality reduction methods suggest to concatenate the datasets into a single vector space. However, this methodology is sensitive to scalings of each data component. It does not utilize for example the fact that noise in both datasets could be uncorrelated. It assumes that both datasets lie in one high dimensional space which is not always true.

The problem of learning from two views has been studied in the field of spectral clustering. Most of these studies have been focused on classification and clustering that are based on spectral characteristics of the data while using two or more sampled views. Some approaches, which address this problem, are Bilinear Model [9], Partial Least Squares [10] and Canonical Correlation Analysis [11]. These methods are powerful for learning the relation between different views but do not provide separate insights or combined into the low dimensional geometry or structure of each view. Recently, a few kernel based methods (e.g [12]) propose a model of co-regularizing kernels in both views in a way that resembles joint diagonalization. It is done by searching for an orthogonal transformation that maximizes the diagonal terms of the kernel matrices obtained from all views. A penalty term, which incorporates the disagreement between clusters from the views, was added. Their algorithm is based on alternating maximization procedure. A mixture of Markov chains is proposed in [13] to model multiple views in order to apply spectral clustering. It deals with two cases in graph theory: directed and undirected graph where the second case is related to our work. This approach converges the undirected graph problem to a Markov chains averaging where each is constructed separately within the views. A way to incorporate a given multiple metrics for the same data using a cross diffusion process is described in [14]. They define a new diffusion distance which is useful for classification, clustering or retrieval tasks. However, the proposed process is not symmetrical thus does not allow to compute an embedding. An iterative algorithm for spectral clustering is proposed in [15]. The idea is to iteratively modify each view using the representation of the other view. The problem of two manifolds, which were derived from the same data (i.e two views), is described in [16]. This approach is similar to Canonical Correlation Analysis [17] that seeks a linear transformation that maximizes the correlation among the views. It demonstrates the power of this method in canceling uncorrelated noise present in both views. Furthermore, [16] applies its method to a non-linear system identification task. A similar approach is proposed in [18]. It suggests data modeling that uses a bipartite graph and then, based on the ‘minimum-disagreement’ algorithm, partitions the dataset. This approach attempts to minimize the cluster’s disagreement between two views. The study presented in [19] utilizes the agreement also called consensus between different views to extract the geometric information from all views. The framework takes advantage of properties of the Mahalanobis distance to compute a robust multi-view kernel.

The problem of multi-view dimensionality reduction was also studied using Gaussian Process Latent Variable Models [20]. The work by [21] uses a Gaussian process regression to learn common hidden structure. The studies by [22] and [23] demonstrate the capabilities of such models for extracting meaningful parameters from images. A related work by [24] attempts to maximize the mutual information between the sampled views and the latent variables. Studies such as [25], [26], [27] use a probabilistic CCA to factorize the data to a common and view specific information.

In this work, we present a framework based on the construction in [18] and show that this approach is a special case of a more general diffusion based process. We build and analyze a new framework that generalizes the random walk model while utilizing multiple views. Our proposed method utilizes the intrinsic relation within each view as well as the mutual relations between views. The multi-view is achieved by defining a cross diffusion process in which a special structured random walk is imposed between the various views. Within this framework, new diffusion distances are defined to analyze the spectra of the new kernels and compute the infinitesimal generator to where the multi-view-based our kernel converges. The constructed multi-view kernel matrix is similar to a symmetric matrix thus guarantees real eigenvalues and eigenvectors, this property enables us to define an multi-view embedding. The advantages of the proposed method for manifold learning and spectral clustering are explored using vast experiments.

The paper has the following structure: Background is given in section II. Section III presents and analyzes the multi-view framework. Section IV-F studies the asymptotic properties of the proposed kernel $\widehat{\mathbf{K}}$ (Eq. (5)). Section VI presents the experimental results.

II. BACKGROUND

A. General dimensionality reduction framework

Consider a high dimensional dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_M\} \in \mathbb{R}^{M \times N}$, $\mathbf{x}_i \in \mathbb{R}^N$, $i = 1, \dots, M$. The goal is to find a low dimensional representation $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \dots, \mathbf{z}_M\} \in \mathbb{R}^{M \times S}$, $\mathbf{z}_i \in \mathbb{R}^S$, $i = 1, \dots, M$, such that $S \ll N$ and the local connectivities among the multidimensional data points are preserved. This problem setup is based on the assumption that the data is represented (viewed) by a single vector space (single view).

B. Diffusion Maps (DM)

DM [5] is a dimensionality reduction method that finds the intrinsic geometry in the data. This framework is highly effective when the data is densely sampled from some low dimensional manifold that is not linear. Given a high dimensional dataset \mathbf{X} , the DM framework contains the following steps:

- 1) A kernel function $\mathcal{K} : \mathbf{X} \times \mathbf{X} \rightarrow \mathbb{R}$ is chosen. It is represented by a matrix $\mathbf{K} \in \mathbb{R}^{M \times M}$ which satisfies for all $(\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{X}$ the following properties: Symmetry: $K_{i,j} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{K}(\mathbf{x}_j, \mathbf{x}_i)$, positive semi-definiteness: $\mathbf{v}_i^T \mathbf{K} \mathbf{v}_i \geq 0$ for all $\mathbf{v}_i \in \mathbb{R}^M$ and $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \geq 0$. These properties guarantee that the matrix \mathbf{K} has real eigenvectors and non-negative real eigenvalues. Gaussian kernel is a common example where $K_{i,j} = \exp\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_x^2}\}$ with an L_2 norm as the affinity measure between two data vectors;
- 2) By normalizing the kernel using \mathbf{D} where $D_{i,i} = \sum_j K_{i,j}$, we compute the following matrix elements:

$$P_{i,j}^x = \mathcal{P}(\mathbf{x}_i, \mathbf{x}_j) = [\mathbf{D}^{-1} \mathbf{K}]_{i,j}. \quad (1)$$

The resulting matrix $\mathbf{P}^x \in \mathbb{R}^{M \times M}$ can be viewed as the transition kernel of a (fictitious) Markov chain on \mathbf{X} such that the expression $[(\mathbf{P}^x)^t]_{i,j} = p_t(\mathbf{x}_i, \mathbf{x}_j)$ describes the transition probability from point \mathbf{x}_i to point \mathbf{x}_j in t steps.

- 3) Spectral decomposition is applied to matrix \mathbf{P}^x or to one of its powers $(\mathbf{P}^x)^t$ to obtain a sequence of eigenvalues $\{\lambda_m\}$ and normalized eigenvectors $\{\psi_m\}$ that satisfies $\mathbf{P}^x \psi_m = \lambda_m \psi_m$, $m = 0, \dots, M-1$;
- 4) Define a new representation for the dataset \mathbf{X}

$$\Psi_t(\mathbf{x}_i) : \mathbf{x}_i \mapsto [\lambda_1^t \psi_1[i], \lambda_2^t \psi_2[i], \lambda_3^t \psi_3[i], \dots, \lambda_{M-1}^t \psi_{M-1}[i]]^T \in \mathbb{R}^{M-1}, \quad (2)$$

where t is the selected number of steps and $\psi_m[i]$ denotes the i^{th} element of ψ_m .

The main idea behind this representation is that the Euclidian distance between two data points in the new representation is equal to the weighted L_2 distance between the conditional probabilities $p_t(\mathbf{x}_i, \cdot)$, and $p_t(\mathbf{x}_j, \cdot)$, $i, j = 1, \dots, M$ (the i -th and j -th rows of \mathbf{P}^t). The following is referred as the Diffusion Distance

$$\mathcal{D}_t^2(\mathbf{x}_i, \mathbf{x}_j) = \|\Psi_t(\mathbf{x}_i) - \Psi_t(\mathbf{x}_j)\|^2 = \sum_{m \geq 1} \lambda_m^{2t} (\psi_m[i] - \psi_m[j])^2 = \|p_t(\mathbf{x}_i, \cdot) - p_t(\mathbf{x}_j, \cdot)\|_{\mathbf{W}^{-1}}^2, \quad (3)$$

where \mathbf{W} is a diagonal matrix with elements $W_{i,i} = \frac{D_{i,i}}{\sum_{i=1}^M D_{i,i}}$. This equality is proven in [5].

- 5) The desired accuracy $\delta \geq 0$ is chosen for the diffusion distance defined by Eq. (3) such that $s(\delta, t) = \max\{\ell \in \mathbb{N} \text{ such that } |\lambda_\ell|^t > \delta |\lambda_1|^t\}$. By using δ , a new mapping of $s(\delta, t)$ dimensions is defined as

$$\Psi_t^{(\delta)} : X \rightarrow [\lambda_1^t \psi_1[i], \lambda_2^t \psi_2[i], \lambda_3^t \psi_3[i], \dots, \lambda_s^t \psi_s[i]]^T \in \mathbb{R}^{s(\delta, t)}.$$

This approach has been found useful in various fields. As previously noted, it is limited to a single view representation. A common extension of this approach to multiple views is to use a data concatenation from all views and then apply the diffusion framework. This method assumes orthogonality of the sampled dimensions which is an unrealistic assumption in many cases. Furthermore, this approach can create

redundancy in some dimensions and requires scaling of each dimension separately such that none is preferable over the others. Previous studies such as [28], [29] apply the DM framework to each view individually and then incorporated the learned mapping from various views. However, they do not exploit the mutual relations which might exist between the different views to create and utilize the correct mapping.

III. MULTI-VIEW DIMENSIONALITY REDUCTION

Problem Formulation: Given multiple sets of observations $\mathbf{X}^l, l = 1, \dots, L$. Each view is a high dimensional dataset $\mathbf{X}^l = \{\mathbf{x}_1^l, \mathbf{x}_2^l, \mathbf{x}_3^l, \dots, \mathbf{x}_M^l\} \in \mathbb{R}^{M \times N_l}$, N_l is the dimension of each feature space. Note that a bijective correspondence between views is assumed. For each view $l = 1, \dots, L$, we seek for a lower dimensional representation that preserves the interactions between multidimensional data points within a given view \mathbf{X}^l and among the views $\{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^L\}$.

A. Multi-view Diffusion Maps

We begin by generalizing the DM framework for handling a multi-view scenario. Our goal is to impose a random walk model using the local connectivities between data points within all views. Our way to generalize the DM framework is by restraining the random walker to “hop” between views in each step. The construction requires to choose symmetrical positive semi-definite kernels for each view $\mathcal{K}^l : \mathbf{X}^l \times \mathbf{X}^l \rightarrow \mathbb{R}, l = 1, \dots, L$, we use the Gaussian function

$$K_{i,j}^l = \exp\left\{-\frac{\|\mathbf{x}_i^l - \mathbf{x}_j^l\|^2}{2\sigma_l^2}\right\}, \quad (4)$$

then the multi-view kernel is formed by the following matrix

$$\widehat{\mathbf{K}} = \begin{bmatrix} \mathbf{0}_{M \times M} & \mathbf{K}^1 \mathbf{K}^2 & \mathbf{K}^1 \mathbf{K}^3 & \dots & \mathbf{K}^1 \mathbf{K}^L \\ \mathbf{K}^2 \mathbf{K}^1 & \mathbf{0}_{M \times M} & \mathbf{K}^2 \mathbf{K}^3 & \dots & \mathbf{K}^2 \mathbf{K}^L \\ \mathbf{K}^3 \mathbf{K}^1 & \mathbf{K}^3 \mathbf{K}^2 & \mathbf{0}_{M \times M} & \dots & \mathbf{K}^3 \mathbf{K}^L \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \mathbf{K}^L \mathbf{K}^1 & \mathbf{K}^L \mathbf{K}^2 & \mathbf{K}^L \mathbf{K}^3 & \dots & \mathbf{0}_{M \times M} \end{bmatrix}. \quad (5)$$

Finally, by using the diagonal matrix $\widehat{\mathbf{D}}$ where $\widehat{D}_{i,i} = \sum_j \widehat{K}_{i,j}$, the normalized row-stochastic matrix is defined as

$$\widehat{\mathbf{P}} = \widehat{\mathbf{D}}^{-1} \widehat{\mathbf{K}}, \quad \widehat{P}_{i,j} = \frac{\widehat{K}_{i,j}}{\widehat{D}_{i,i}}, \quad (6)$$

where the m, l block is a square $M \times M$ matrix located at $[1 + (m-1)M, 1 + (l-1)M], l = 1, \dots, L$. This block describes the probability of transition between view \mathbf{X}^m and \mathbf{X}^l .

B. Alternative multi-view approaches

In this section, we describe two additional methods for incorporating different views. We do not analyze these approaches but use them as references for comparisons in the experimental evaluations.

1. **Kernel Product DM (KP):** The kernel matrix elements are multiplied element wise $\mathbf{K}^\circ \triangleq \mathbf{K}^1 \circ \mathbf{K}^2 \circ \dots \circ \mathbf{K}^L$, $K_{i,j}^\circ \triangleq K_{i,j}^1 \cdot K_{i,j}^2 \cdot \dots \cdot K_{i,j}^L$. Then, they are normalized by the sum of rows. The resulting row stochastic matrix is

$$P_{i,j}^\circ = [\mathbf{D}^{\circ-1} \mathbf{K}^\circ]_{i,j}, \quad (7)$$

where $D_{i,i}^\circ = \sum_j K_{i,j}^\circ$.

Lema 1. In the special case of a Gaussian kernel with $\sigma_1 = \sigma_2 = \dots = \sigma_L$ in Eq. (4), the resulting matrix \mathbf{K}° is equal to the matrix \mathbf{K}^w constructed using the concatenated vector $\mathbf{w}_i = [(\mathbf{x}_i^1)^T, \dots, (\mathbf{x}_i^L)^T]^T$ such that $K_{i,j}^w = \exp\{-\frac{\|\mathbf{w}_i - \mathbf{w}_j\|^2}{2\sigma_w^2}\}$. The scale is set to

$$\sigma_w = \sqrt{\sum_{i=1}^L \sigma_i^2} = \sqrt{L \cdot \sigma_1^2}, \quad (8)$$

where the last equality holds only for this special case.

This approach, which corresponds to [5], will be referred as the Kernel Product DM in section VI.

2. Kernel Sum DM (KS): The sum kernel is defined as $\mathbf{K}^+ \triangleq \sum_{l=1}^L \mathbf{K}^l$. By normalizing the sum kernel by the diagonal matrix $\mathbf{D}^+ = \sum_j K_{i,j}^+$, we get

$$P_{i,j}^+ = [\mathbf{D}^{+^{-1}} \mathbf{K}^+]_{i,j}. \quad (9)$$

This random walk sums the step probabilities from each view. This approach is proposed in [13].

C. Probabilistic interpretation of $\hat{\mathbf{P}}^t$

In our proposed construction (Eqs. (4), (5) and (6)), the entries $[\hat{\mathbf{P}}^t]_{i,j} = \hat{p}_t(\mathbf{x}_i^1, \mathbf{x}_j^1)$ denote for each $i, j = 1, \dots, M$, the transition probability from node \mathbf{x}_i^1 to node \mathbf{x}_j^1 in t time steps by “hopping” between the views \mathbf{X}^1 and $\mathbf{X}^l, l = 2, \dots, L$ in each time step, where L is the total number of views. Note that due to the block-anti-diagonal structure of $\widehat{\mathbf{K}}$ (and $\widehat{\mathbf{P}}$ (Eq. (6))), this probability is zero for $t = 1$. However, for higher values of t , this probability is nonzero describing a time transition from view \mathbf{X}^1 through any view $\mathbf{X}^l, l = 2, \dots, L$ and back to \mathbf{X}^1 . In the same way, $[\hat{\mathbf{P}}^t]_{i+(l-1) \cdot M, j+(l-1)M} = \hat{p}_t(\mathbf{x}_i^l, \mathbf{x}_j^l)$ denotes the transition probability from node \mathbf{x}_i^l to node $\mathbf{x}_j^l, i, j = 1, \dots, M$, in t time steps. Likewise, $[\hat{\mathbf{P}}^t]_{i+(l-1) \cdot M, j+(m-1)M} = \hat{p}_t(\mathbf{x}_i^l, \mathbf{x}_j^m)$ denotes the transition probability from node \mathbf{x}_i^l to node $\mathbf{x}_j^m, i, j = 1, \dots, M, l \neq m$ in t time steps.

1) *Smoothing effect $t = 1$:* For simplicity let us examine the term $\hat{p}_t(\mathbf{x}_i^l, \mathbf{x}_j^m), l \neq m$ for $t = 1$. The transition probability for $t = 1$ is

$$\hat{p}_1(\mathbf{x}_i^l, \mathbf{x}_j^m) = \frac{\sum_s K_{i,s}^l K_{s,j}^m}{\hat{D}_{i,i}}.$$

This probability takes into consideration all the various connectivities of node \mathbf{x}_i^l to node \mathbf{x}_s^l and the connectivities of the corresponding node \mathbf{x}_s^m to the destination node \mathbf{x}_j^m . The proposed multi-view approach has a smoothing effect in terms of the transition probability. By smoothing effect, we mean that the probability of transitioning from \mathbf{x}_i^l to \mathbf{x}_j^m could be larger than zero even if $K_{i,j}^l = 0$ and $K_{i,j}^m = 0$. Assume that there is a subset $\mathcal{S} = \{s_1, \dots, s_F\}$ such that $K_{i,s_f}^l > 0$ and $K_{s_f,j}^m > 0, f = 1, \dots, F$, by definition of the multi-view probability we get that $\hat{p}_1(\mathbf{x}_i^l, \mathbf{x}_j^m) > 0$.

Figure 1 illustrates the multi-view transition probabilities compared to a single view approach using two deformed Swiss-roll manifolds ($L = 2$). In each view, there is no probability of transition from one side of the gap to the other. The multi-view transition probability is non-zero for points at both sides of the gap. This smoothing effect occurs because the gap is located near different points on both views, thus allowing the multi-view kernel to smooth the nonlinear gap.

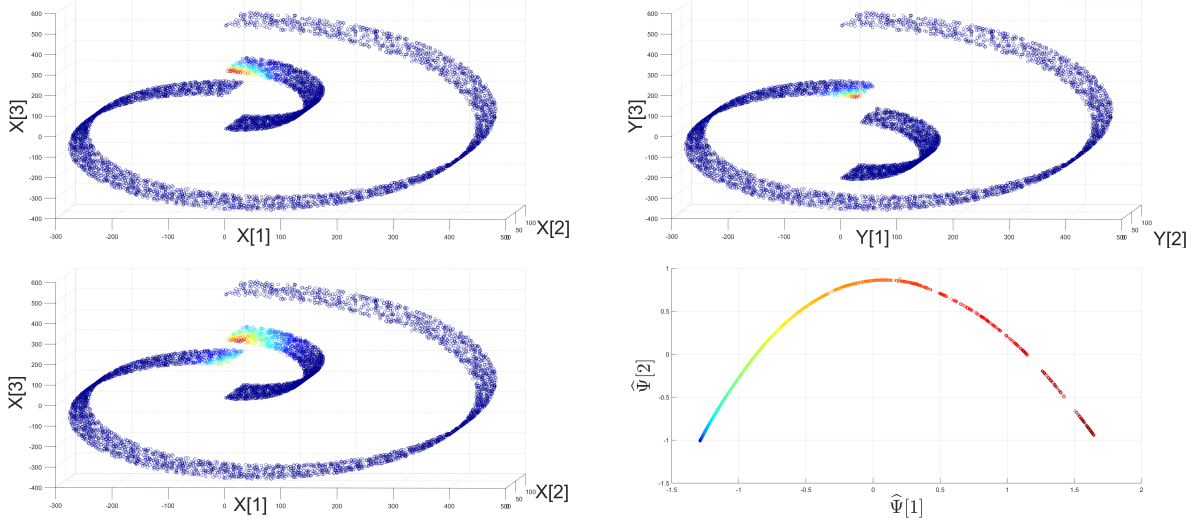


Fig. 1: Top left: Non-smooth Swiss Roll sampled from View-I (\mathbf{X}), colored by the single view probability of transition ($t = 1$) from \mathbf{x}_1 to \mathbf{x}_\cdot . Top right: second Swiss Roll sampled from View-II (\mathbf{Y}), colored by the single view probability of transition ($t = 1$) from \mathbf{y}_1 to \mathbf{y}_\cdot . Bottom left: the first Swiss Roll colored the multi-view probabilities of transition ($t = 1$) from \mathbf{x}_i to \mathbf{y}_\cdot . This point \mathbf{x}_1 denoted with an arrow in the top left figure. Bottom right: a low dimensional representation extracted based on the multi-view transition matrix $\hat{\mathbf{P}}$ (Eq. 6).

2) *Increasing the diffusion step t* : Under the stochastic Markov model assumption, increasing the power of the matrix $\hat{\mathbf{P}}$ spreads the probability along the data points based on the connectivities in all views. This probability spread as describe in [5] reduces the influence of high eigenvectors on the diffusion distance (Eq. (10)). This implies that the eigenvectors corresponding to low eigenvalues have a low-frequency content, whereas the eigenvectors corresponding to the high eigenvalues describe the oscillatory behavior of the data [5]. In Fig. 2, we present the eigenvalues of the matrix $\hat{\mathbf{P}}^t$ at different values of t . For the experiment we have generated $L = 3$ Swiss rolls with $M = 1200$ data points each. It is evident that the numerical rank of $\hat{\mathbf{P}}^t$ decreases for higher values of t .

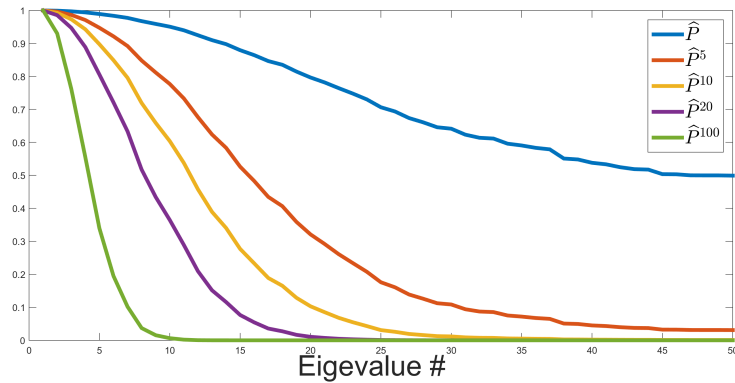


Fig. 2: The decay of the eigenvalues for increasing powers of the matrix $\hat{\mathbf{P}}$.

D. Multi-view diffusion distance

In a variety of real data types, the Euclidean distance does not provide a sufficient information about the intrinsic relations between data points. The Euclidean distance is highly sensitive to scaling and rotations of multidimensional data points. Tasks such as classification, clustering or system identification require a measure for the intrinsic connectivity between data points. This type of measure is only satisfied locally by the Euclidean distance in the high dimensional ambient space. The multi-view diffusion kernel (defined in section (III-A)) describes all the small local connections between data points. The row stochastic matrix $\hat{\mathbf{P}}^t$ (Eq. (6)) incorporates all the possibilities for having a transition in t time steps between data points that are hopping between both views. For a fixed value $t > 0$, two data points are *intrinsically similar* if the conditional distributions $\hat{\mathbf{p}}_t(\mathbf{x}_i, :) = [\hat{\mathbf{P}}^t]_{i,:}$ and $\hat{\mathbf{p}}_t(\mathbf{x}_j, :) = [\hat{\mathbf{P}}^t]_{j,:}$ are similar. This type of similarity measure indicates that the points \mathbf{x}_i and \mathbf{x}_j are similarly connected to several mutual points. Thus, they are connected by a geometrical path. In many cases, a small Euclidean distance can be misleading due to the fact that two data points can be “close” without having any geodesic path that connects them. Comparing the transition probabilities is more robust as it takes into consideration all of the local connectivities between the compared points. Therefor, even if two points do not have a small Euclidean distance between them, they may have many common neighbors and thus have a low diffusion distance.

Based on this observation, by expanding the single view construction given in [5], we define the weighted inner view diffusion distances for the first view as

$$\mathcal{D}_t^2(\mathbf{x}_i^1, \mathbf{x}_j^1) \triangleq \sum_{k=1}^{L \cdot M} \frac{([\hat{\mathbf{P}}^t]_{i,k} - [\hat{\mathbf{P}}^t]_{j,k})^2}{\phi_o(k)} = \|(\mathbf{e}_i - \mathbf{e}_j)^T \hat{\mathbf{P}}^t\|_{\hat{\mathbf{D}}^{-1}}^2, \quad (10)$$

where $1 \leq i, j \leq M$, \mathbf{e}_i is the i -th column of an $L \cdot M \times L \cdot M$ identity matrix, ϕ_o is the first left eigenvector of $\hat{\mathbf{P}}$ and its k -th element is $\phi_o(k) = \hat{D}_{k,k}$. The weighted norm of \mathbf{x} is defined by $\|\mathbf{x}\|_{\mathbf{W}}^2 = \mathbf{x}^T \mathbf{W} \mathbf{x}$. Similarly, the weighted norm is defined for the l -th view

$$\mathcal{D}_t^2(\mathbf{x}_i^l, \mathbf{x}_j^l) \triangleq \sum_{k=1}^{L \cdot M} \frac{([\hat{\mathbf{P}}^t]_{\tilde{l}+i,k} - [\hat{\mathbf{P}}^t]_{\tilde{l}+j,k})^2}{\phi_o(k)} = \|(\mathbf{e}_{\tilde{l}+i} - \mathbf{e}_{\tilde{l}+j})^T \hat{\mathbf{P}}^t\|_{\hat{\mathbf{D}}^{-1}}^2, \quad (11)$$

where $\tilde{l} = (l-1) \cdot M$. The main advantage of these distances (Eqs. (10) and (11)) is that they can be expressed in terms of the eigenfunctions and the eigenvectors of the matrix $\hat{\mathbf{P}}$. This insight allows us to use a representation (defined in section III-E) where the induced Euclidean distance is proportional to the diffusion distances defined in Eqs. (10) and (11).

Theorem 1. *The inner view diffusion distance defined by Eqs. (10) and (11) is equal to*

$$\mathcal{D}_t^2(\mathbf{x}_i^l, \mathbf{x}_j^l) = \sum_{k=1}^{L \cdot M - 1} \lambda_k^{2t} (\psi_k[\tilde{l} + i] - \psi_k[\tilde{l} + j])^2, \quad i, j = 1, \dots, M, \quad (12)$$

where $\tilde{l} = (l-1) \cdot M$.

Proof. We express $\hat{\mathbf{P}}^t \hat{\mathbf{D}}^{-1} (\hat{\mathbf{P}}^t)^T$ by $\hat{\mathbf{P}}^t \hat{\mathbf{D}}^{-1} (\hat{\mathbf{P}}^t)^T = \Psi \Lambda^t \Phi^T \hat{\mathbf{D}}^{-1} \Phi \Lambda^t \Psi^T = \Psi \Lambda^{2t} \Psi^T$ since $\Phi^T \hat{\mathbf{D}}^{-1} \Phi = \Pi^T \Pi = \mathbf{I}$. Therefore, $\mathcal{D}_t^2(\mathbf{x}_i^l, \mathbf{x}_j^l) =$

$$\begin{aligned} & \|(\mathbf{e}_{\tilde{l}+i} - \mathbf{e}_{\tilde{l}+j})^T \hat{\mathbf{P}}^t\|_{\hat{\mathbf{D}}^{-1}}^2 = (\mathbf{e}_{\tilde{l}+i} - \mathbf{e}_{\tilde{l}+j})^T \hat{\mathbf{P}}^t \hat{\mathbf{D}}^{-1} (\hat{\mathbf{P}}^t)^T (\mathbf{e}_{\tilde{l}+i} - \mathbf{e}_{\tilde{l}+j}) = \\ & (\mathbf{e}_{\tilde{l}+i} - \mathbf{e}_{\tilde{l}+j})^T \Psi \Lambda^{2t} \Psi^T (\mathbf{e}_{\tilde{l}+i} - \mathbf{e}_{\tilde{l}+j}) = \sum_{k=0}^{L \cdot M - 1} \lambda_k^{2t} (\psi_k[\tilde{l} + i] - \psi_k[\tilde{l} + j])^2 = \\ & \sum_{k=1}^{L \cdot M - 1} \lambda_k^{2t} (\psi_k[\tilde{l} + i] - \psi_k[\tilde{l} + j])^2. \end{aligned}$$

$\ell = 0$ is excluded due to $\Psi_0 = \mathbf{1}$ (an all-ones vector) that holds for all stochastic matrices. \square

E. Multi-view data parametrization

Tasks such as classification, clustering or regression in a high-dimension feature space are considered to be computationally expensive. In addition, the performance of these tasks is highly dependent on the distance measure. As explained in section III-D, distance measures in the original ambient space are meaningless in many real life situations. Interpreting Theorem 1 in terms of Euclidean distance enables us to define mappings for every view $\mathbf{X}^l, l = 1, \dots, L$, using the right eigenvectors of $\hat{\mathbf{P}}$ (Eq. (6)) weighted by λ_i^l . The representation for instances in \mathbf{X}^l is given by

$$\hat{\Psi}_t(\mathbf{x}_i^l) : \mathbf{x}_i^l \mapsto [\lambda_1^l \psi_1[i + \bar{l}], \dots, \lambda_{M-1}^l \psi_{M-1}[i + \bar{l}]]^T \in \mathbb{R}^{M-1}, \quad (13)$$

where $\bar{l} = (l - 1) \cdot M$. These L mappings capture the intrinsic geometry of the views as well as the mutual relation between them. As shown in [30], the set of eigenvalues λ_m has a decaying property such that $1 = |\lambda_0| \geq |\lambda_1| \geq \dots \geq |\lambda_{M-1}|$. Exploiting the decaying property enables us to represent data up to a dimension r where $r \ll N_1, \dots, N_L$. The dimension $r \equiv r(\delta)$ is determined by approximating the diffusion distance (Eq. (12)) up to a desired accuracy δ . This argument is expanded in section IV-D. The reduced dimension version of $\hat{\Psi}_t(\mathbf{X})$ is denoted by $\hat{\Psi}_t^r(\mathbf{X})$.

Using the inner view diffusion distances defined in Eqs. (10) and (11), we define a multi-view diffusion distance as a linear combination of the inner views distances such that

$$\mathcal{D}_t^{(MV)^2}(i, j) \triangleq \sum_{l=1}^L \|\hat{\Psi}_t^r(\mathbf{x}_i^l) - \hat{\Psi}_t^r(\mathbf{x}_j^l)\|^2. \quad (14)$$

This distance is the induced Euclidean distance in a space constructed from the concatenation of all low dimensional multi-view mappings

$$\hat{\Psi}_t(\mathbf{X}) = [\hat{\Psi}_t^r(\mathbf{X}^1), \hat{\Psi}_t^r(\mathbf{X}^2), \dots, \hat{\Psi}_t^r(\mathbf{X}^L)]. \quad (15)$$

This mapping is used in section VI-B for the experimental evaluation of clustering.

F. Multi-view kernel bandwidth

When constructing the Gaussian kernels $\mathbf{K}^l, l = 1, \dots, L$, in Eq. (4), the values of the scale (width) parameter σ_l^2 have to be set. Setting these values to be too small may result in very small local neighborhoods that are unable to capture the local structure around the data point. On the contrary, setting the values to be too large may result in a fully connected graph that may generate a coarse description of the data. In [31], a max-min measure is suggested such that the scale becomes

$$\sigma_l^2 = C \cdot \max_j [\min_{i, i \neq j} (\|\mathbf{x}_i^l - \mathbf{x}_j^l\|^2)], \quad (16)$$

where C is set within the range $[1, 1.5]$. This approach attempts to set a small scale to maintain local connectivities. This single view approach could be relaxed in the multi-view scenario. The multi-view kernel $\hat{\mathbf{K}}$ (Eq. 5) contains multiplication of single view kernel matrices $\mathbf{K}^l, l = 1, \dots, L$ (Eq. 4). The diagonal values of each kernel matrix $\mathbf{K}^l, l = 1, \dots, L$ are all 1's, therefore, a connectivity in only one view is sufficient. This insight suggests that smaller value for the parameter C could be used.

Another scheme by [32] aims to find a range of values for σ_l . The idea is to compute the kernel \mathbf{K}^l (Eq. (4)) for various values of σ and search for the range of values where the Gaussian bell shape exists. The range is identified by applying a logarithmic function to the sum of the kernel. We expand this idea for a multi-view scenario based on the following algorithm:

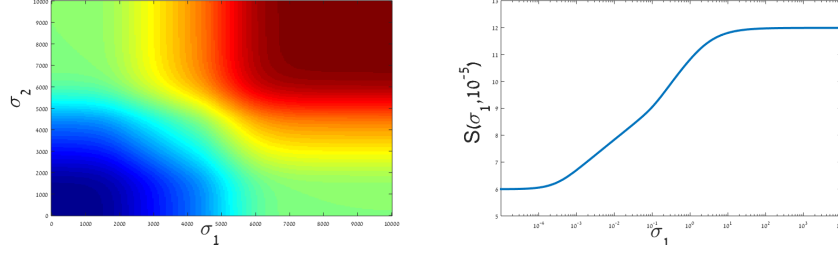


Fig. 3: Left: an example of the two dimensional function $S(\sigma_l, \sigma_m)$. Right: a slice at the first row ($\sigma_2 = 10^{-5}$). The asymptotes are clearly visible in both figures. Algorithm 1 exploits the multi-view to set a small scale parameter for both views.

Algorithm 1 Multi-view kernel bandwidth selection

Input: Multiple sets of observations (views) $\mathbf{X}^l, l = 1, \dots, L$.

Output: Scale parameters for all views $\{\sigma_1, \dots, \sigma_L\}$.

- 1: Compute Gaussian kernels $\mathbf{K}^l(\sigma_l), l = 1, \dots, L$ for several values of σ_l .
 - 2: Compute for all pairs $l \neq m$: $S^{lm}(\sigma_l, \sigma_m) = \sum_i \sum_j K_{i,j}^{lm}(\sigma_l, \sigma_m)$, where $\mathbf{K}^{lm}(\sigma_l, \sigma_m) = \mathbf{K}^l(\sigma_l) \cdot \mathbf{K}^m(\sigma_m)$.
 - 3: **for** $l = 1 : L$ **do**
 - 4: Find the minimal value for σ_l such that $S^{lm}(\sigma_l, \sigma_m)$ is linear for all $m \neq l$.
 - 5: **end for**
-

Note that the two dimensional function $S^{lm}(\sigma_l, \sigma_m)$ consists of two asymptotes, $S^{lm}(\sigma_l, \sigma_m) \xrightarrow{\sigma_l, \sigma_m \rightarrow 0} \log(N)$, and $S^{lm}(\sigma_l, \sigma_m) \xrightarrow{\sigma_l, \sigma_m \rightarrow \infty} \log(N^3) = 3\log(N)$, since for $\sigma_l, \sigma_m \rightarrow 0$, both \mathbf{K}^l and \mathbf{K}^m approach the Identity matrix, and for $\sigma_l, \sigma_m \rightarrow \infty$, both \mathbf{K}^l and \mathbf{K}^m approach all-ones matrices. An example of the plot $S^{lm}(\sigma_l, \sigma_m)$ for two views ($L = 2$) is presented in Fig. 3.

IV. COUPLED VIEWS $L = 2$

In this section we provide analytical results for the simple case of a coupled data set (i.e $L = 2$). Some of the results could be expanded to a larger number of views but not in a straight forward manner. To simplify the notation in the rest of this section we denote $\mathbf{X} \triangleq \mathbf{X}^1$ and $\mathbf{Y} \triangleq \mathbf{X}^2$.

A. Coupled mapping

The mappings provided by our approach (Eq. (13)) are justified by the relations given by Eq. (12). In this section, we provide another analytic justification for the proposed mapping. We begin with an analysis of a 1-dimensional mapping for each view. Let $\rho(\mathbf{x}) = (\rho(\mathbf{x}_1), \rho(\mathbf{x}_2), \dots, \rho(\mathbf{x}_M))$ and $\rho(\mathbf{y}) = (\rho(\mathbf{y}_1), \rho(\mathbf{y}_2), \dots, \rho(\mathbf{y}_M))$ denote such mappings (one for each view) and let $\hat{\rho} \triangleq (\rho(\mathbf{x}), \rho(\mathbf{y}))$ and $\hat{\rho}_i \triangleq (\rho(\mathbf{x}_i), \rho(\mathbf{y}_i))$. Define $\mathbf{K}^z = \mathbf{K}^x \cdot \mathbf{K}^y$, where $\mathbf{K}^x, \mathbf{K}^y$ are computed based on Eq. (4). Our mapping should preserve local connectivities, therefore, we want to ensure that if the data points i and j are close in both views, then $\hat{\rho}_i$ and $\hat{\rho}_j$ will be close. Minimization of the objective function

$$\underset{\hat{\rho}}{\operatorname{argmin}} \sum_{i,j} \left[(\rho(\mathbf{x}_i) - \rho(\mathbf{x}_j))^2 K_{i,j}^z + (\rho(\mathbf{y}_i) - \rho(\mathbf{y}_j))^2 (K_{i,j}^z)^T \right], \quad (17)$$

with additional constraints provides such a connectivity preserving mapping. If $K_{i,j}^z$ is small indicating a low connectivity between data point i and j , the distance between $\hat{\rho}_i$ and $\hat{\rho}_j$ can be large. On the other

hand, if $K_{i,j}^z$ is large indicating a high connectivity between point i and j , the distance between $\hat{\rho}_i$ and $\hat{\rho}_j$ will be small to minimize the objective function.

Theorem 2. *Setting $\hat{\rho} = \psi_1$ minimizes the objective function in Eq. (17), where ψ_1 is the second eigenvector of the eigenvalue problem $\lambda_i \hat{\mathbf{P}} = \psi_i \hat{\mathbf{P}}$.*

Proof.

$$\begin{aligned} & \sum_{i,j} \left[(\rho(\mathbf{x}_i) - \rho(\mathbf{x}_j))^2 K_{i,j}^z + (\rho(\mathbf{y}_i) - \rho(\mathbf{y}_j))^2 K_{i,j}^z \right] = \sum_{i,j} \rho(\mathbf{x}_i)^2 K_{i,j}^z + \sum_{i,j} \rho(\mathbf{x}_j)^2 (K_{i,j}^z)^T \\ & - \sum_{i,j} 2\rho(\mathbf{x}_i)\rho(\mathbf{x}_j)K_{i,j}^z + \sum_{i,j} \rho(\mathbf{y}_i)^2 K_{i,j}^z + \sum_{i,j} \rho(\mathbf{y}_j)^2 (K_{i,j}^z)^T - \sum_{i,j} 2\rho(\mathbf{y}_i)\rho(\mathbf{y}_j)K_{i,j}^z = \\ & \sum_i \rho(\mathbf{x}_i)^2 D_{i,i}^{rows} + \sum_j \rho(\mathbf{x}_j)^2 D_{j,j}^{cols} - \sum_{i,j} 2\rho(\mathbf{y}_i)\rho(\mathbf{y}_j)K_{i,j}^z + \\ & \sum_i \rho(\mathbf{y}_i)^2 D_{i,i}^{cols} + \sum_j \rho(\mathbf{y}_j)^2 D_{j,j}^{rows} - \sum_{i,j} 2\rho(\mathbf{y}_i)\rho(\mathbf{y}_j)K_{i,j}^z = \\ & \begin{bmatrix} \rho(x) & \rho(y) \end{bmatrix} \left[\begin{bmatrix} D^{rows} & \mathbf{0}_{M \times M} \\ \mathbf{0}_{M \times M} & D^{cols} \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{M \times M} & \mathbf{K}^z \\ (\mathbf{K}^z)^T & \mathbf{0}_{M \times M} \end{bmatrix} \right] \begin{bmatrix} \rho(x)^T \\ \rho(y)^T \end{bmatrix}. \end{aligned}$$

By adding a scaling constrain, the minimization problem is rewritten as

$$\begin{aligned} & \underset{\hat{\rho} \hat{\mathbf{D}} \hat{\rho}^T = 1}{\operatorname{argmin}} \hat{\rho}(\hat{\mathbf{D}} - \hat{\mathbf{K}})\hat{\rho}^T. \end{aligned} \quad (18)$$

This minimization problem can be solved by finding the minimal eigenvalue of $(\hat{\mathbf{D}} - \hat{\mathbf{K}})\hat{\rho}^T = \bar{\lambda} \hat{\mathbf{D}} \hat{\rho}^T$ since the minimization term is $\rho \bar{\lambda} \hat{\mathbf{D}} \hat{\rho}^T = \bar{\lambda}$. This eigenproblem has a trivial solution which is an eigenvector of all ones (denoted as $\mathbf{1}$) with $\bar{\lambda} = 0$. The following constraint $\hat{\rho} \hat{\mathbf{D}} \mathbf{1} = 0$ was added to remove the trivial solution. The solution is given by the smallest non-zero eigenvalue. Multiplying Eq. (18) by $\hat{\mathbf{D}}^{-1}$ reduces the problem to $\hat{\rho} \hat{\mathbf{P}} = \lambda \hat{\mathbf{P}}$. Thus, we are looking for the eigenvector which corresponds to the second largest eigenvalue. \square

Theorem 2 provides yet another justification to use our proposed mapping Eq. (13).

B. Spectral decomposition

In this section, we show how to efficiently compute the spectral decomposition of $\hat{\mathbf{P}}$ (Eq. (6)) when only two view exist ($L = 2$). The matrix $\hat{\mathbf{P}}$ is algebraically similar to the symmetric matrix $\hat{\mathbf{P}}_s$ where $\hat{\mathbf{P}}_s = \hat{\mathbf{D}}^{1/2} \hat{\mathbf{P}} \hat{\mathbf{D}}^{-1/2} = \hat{\mathbf{D}}^{-1/2} \hat{\mathbf{K}} \hat{\mathbf{D}}^{-1/2}$. Therefore, both $\hat{\mathbf{P}}$ and $\hat{\mathbf{P}}_s$ share the same set of eigenvalues $\{\lambda_m\}$. Due to symmetry of the matrix $\hat{\mathbf{P}}_s$, it has a set of $2M$ real eigenvalues $\{\lambda_i\}_{i=0}^{2M-1} \in \mathbb{R}$ and a corresponding real orthogonal eigenvectors $\{\pi_m\}_{m=0}^{2M-1} \in \mathbb{R}^{2M}$, thus, $\hat{\mathbf{P}}_s = \Pi \Lambda \Pi^T$. By denoting $\Psi = \hat{\mathbf{D}}^{-1/2} \Pi$ and $\Phi = \hat{\mathbf{D}}^{1/2} \Pi$, we conclude that the set $\{\psi_m, \phi_m\}_{m=0}^{2M-1} \in \mathbb{R}^{2M}$ denotes the right and the left eigenvectors of $\hat{\mathbf{P}} = \Psi \Lambda \Phi^T$, respectively, satisfying $\psi_i^T \phi_j = \delta_{i,j}$. In the sequel, we use the symmetric matrix $\hat{\mathbf{P}}_s$ to simplify the analysis.

To avoid the spectral decomposition of a $2M \times 2M$ matrix $\hat{\mathbf{P}}_s$, the spectral decomposition of $\hat{\mathbf{P}}_s$ can be computed using the Singular Value Decomposition (SVD) of the matrix $\bar{\mathbf{K}}^z = \mathbf{D}^{rows-1/2} \mathbf{K}^z \mathbf{D}^{cols-1/2}$ of size $M \times M$ where $D_{i,i}^{rows} = \sum_{j=1}^M K_{i,j}^z$ and $D_{j,j}^{cols} = \sum_{i=1}^M K_{i,j}^z$ are diagonal matrices. Theorem 3 enables us to form the eigenvectors of $\hat{\mathbf{P}}$ as a concatenation of the singular vectors of $\mathbf{K}^z = \mathbf{K}^x \cdot \mathbf{K}^y$.

Theorem 3. *By using the left and right singular vectors of $\mathbf{K}^z = \mathbf{V} \Sigma \mathbf{U}^T$, the eigenvectors and the eigenvalues of $\hat{\mathbf{K}}$ are computed explicitly by*

$$\Pi = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{V} & \mathbf{V} \\ \mathbf{U} & -\mathbf{U} \end{bmatrix}, \Lambda = \begin{bmatrix} \Sigma & \mathbf{0}_{M \times M} \\ \mathbf{0}_{M \times M} & -\Sigma \end{bmatrix}. \quad (19)$$

Proof. Both \mathbf{V} and \mathbf{U} are orthonormal sets, therefore, $\mathbf{u}_i^T \mathbf{u}_j = \delta_{i,j}$, and $\mathbf{v}_i^T \mathbf{v}_j = \delta_{i,j}$, thus, the set $\{\pi_m\}$ is orthonormal. Therefore, $\mathbf{\Pi}\mathbf{\Pi}^T = \mathbf{I}$. By using the construction defined in Eq. (19), $\mathbf{\Pi}\mathbf{\Lambda}\mathbf{\Pi}^T$ is computed explicitly by

$$\mathbf{\Pi}\mathbf{\Lambda}\mathbf{\Pi}^T = \frac{1}{2} \begin{bmatrix} \mathbf{V} & \mathbf{V} \\ \mathbf{U} & -\mathbf{U} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & -\mathbf{\Sigma} \end{bmatrix} \begin{bmatrix} \mathbf{V}^T & \mathbf{U}^T \\ \mathbf{V}^T & -\mathbf{U}^T \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \mathbf{V}\mathbf{\Sigma} & -\mathbf{V}\mathbf{\Sigma} \\ \mathbf{U}\mathbf{\Sigma} & \mathbf{U}\mathbf{\Sigma} \end{bmatrix} \begin{bmatrix} \mathbf{V}^T & \mathbf{U}^T \\ \mathbf{V}^T & -\mathbf{U}^T \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \mathbf{0} & 2\mathbf{K}^z \\ (2\mathbf{K}^z)^T & \mathbf{0} \end{bmatrix} = \widehat{\mathbf{K}}. \text{ The } \mathbf{0} \text{ denotes the } M \times M \text{ matrix of zeros.}$$

□

Thus the proposed mapping in Eq. (13). could be computed for $L = 2$ using the SVD of \mathbf{K}^z , Eq. (19) and $\mathbf{\Psi} = \widehat{\mathbf{D}}^{-1/2}\mathbf{\Pi}$.

C. Cross view diffusion distance

In some physical systems, the observed dataset denoted by \mathbf{X} changes over some underlying parameter denoted by α . Under this model, we can obtain multiple snapshots for various values of α . Each snapshot is denoted by \mathbf{X}^α . If these datasets are high dimensional, quantifying the amount of change in the datasets is a difficult task. This scenario was recently studied in [29]. It generalizes the diffusion framework for cases in which the data changes over the parameter α . An example of such a scenario occurs in hyper-spectral images that change over time. The DM framework is applied in [29] to a fixed value of α . Then, by using the extracted low dimensional mapping, the Euclidean distance enables to quantify the amount of changes over α . This approach is sensitive since every small change in the data can result in different mappings and the mappings are extracted independently. Thus, there is no mutual influence on the extracted mapping. Our approach incorporates the mutual relations of data within the view and the relations among views. This observation enables us to measure in a more robust way the number of variations between two datasets that correspond to a small variation in α . We now define a new diffusion distance. This distance measures the relation between two views, i.e. between all the data points at different values of α . We measure the distance between all the coupled data points among the mappings of the snapshots \mathbf{X}^{α_l} and \mathbf{X}^{α_m} by using the expression

$$\mathcal{D}_t^{(CV)^2}(\mathbf{X}^{\alpha_l}, \mathbf{X}^{\alpha_m}) \triangleq \sum_{i=1}^M \|\widehat{\Psi}_t(\mathbf{x}_i^{\alpha_l}) - \widehat{\Psi}_t(\mathbf{x}_i^{\alpha_m})\|^2. \quad (20)$$

Our kernel matrix is a product of the Gaussian kernel matrices in each view. If these values of the kernel matrices ($\mathbf{K}^{\alpha_l}, \mathbf{K}^{\alpha_m}$) are similar, this corresponds to similarity between the views inner geometry. The right and left singular vectors of the matrix $\mathbf{K}^{\alpha_l} \mathbf{K}^{\alpha_m}$ will be similar, thus, $\mathcal{D}_t^{(CV)}$ will be small.

Theorem 4. *The cross manifold distance (defined in Eq. (20)) is invariant to orthonormal transformations between the ambient spaces \mathbf{X}^{α_l} and \mathbf{X}^{α_m} .*

Proof. Denote an orthonormal transformation matrix $\mathbf{R} : \mathbf{X}^{\alpha_l} \rightarrow \mathbf{X}^{\alpha_m}$ w.l.o.g. by $\mathbf{x}_i^{\alpha_m} = \mathbf{R}\mathbf{x}_i^{\alpha_l}$.

$K_{i,j}^{\alpha_m} = \exp\{-\frac{\|\mathbf{x}_i^{\alpha_m} - \mathbf{x}_j^{\alpha_m}\|^2}{2\sigma_m^2}\} = \exp\{-\frac{\|\mathbf{R}\mathbf{x}_i^{\alpha_l} - \mathbf{R}\mathbf{x}_j^{\alpha_l}\|^2}{2\sigma_l^2}\} = \exp\{-\frac{\|\mathbf{x}_i^{\alpha_l} - \mathbf{x}_j^{\alpha_l}\|^2}{2\sigma_l^2}\} = K_{i,j}^{\alpha_l}$. The last equality is due to the orthonormality of \mathbf{R} and to the choice $\sigma_l = \sigma_m$. Therefore, the matrix $\mathbf{K}^z = (\mathbf{K}^{\alpha_m})^2$ from Eq. (4) is symmetric and its right and left singular vectors are equal, i.e. $\mathbf{U} = \mathbf{V}$, Eq. (19). This induces a repetitive form in $\mathbf{\Psi} = \widehat{\mathbf{D}}^{-1/2}\mathbf{\Pi} \rightarrow \psi_l[i] = \psi_l[M+i]$, $1 \leq i, l \leq M-1 \rightarrow \Psi_t(\mathbf{x}_i^{\alpha_l}) = \Psi_t(\mathbf{x}_i^{\alpha_m})$, thus, $\mathcal{D}_t^{(CM)^2}(\mathbf{X}^{\alpha_l}, \mathbf{X}^{\alpha_m}) = 0$.

□

D. Spectral decay of $\widehat{\mathbf{K}}$

The power of kernel based methods for dimensionality reduction stems from the spectral decay of the kernel's eigenvalues. In this section, we study the relation between the spectral decay of the Kernel Product (Eq. (7)) and our multi-view kernel (Eq. (6)). In section VI-A1, we evaluate the spectral decay empirically using two experiments. The rest of this section is devoted to the theoretical justification for the spectral decay of our proposed framework. We start with some background.

Theorem 5. *The eigenvalues of $\widehat{\mathbf{P}}$ (Eq. (6)) are real and bounded where $|\lambda_i| \leq 1$, $i = 1, \dots, 2M$.*

A similar proof is given in [33].

Proof. As shown in section IV-B, $\widehat{\mathbf{P}}$ is algebraically similar to a symmetric matrix, thus, its eigenvalues are guaranteed to be real. Denote by λ and ψ the eigenvalue and the eigenvector, respectively, such that $\lambda\psi = \widehat{\mathbf{P}}\psi$. Define $i_0 = \underset{1 \leq i \leq 2M}{\operatorname{argmax}} |\psi[i]|$ to be the index of the largest entry in ψ . The maximal value

$\psi(i_0)$ can be computed using $\widehat{\mathbf{P}}$ from Eq. (6) such that $\lambda\psi[i_0] = \sum_{j=1}^{2M} \widehat{P}_{i_0j} \psi[j] \rightarrow |\lambda| = \left| \sum_{j=1}^{2M} \widehat{P}_{i_0j} \frac{\psi[j]}{\psi[i_0]} \right| \leq \sum_{j=1}^{2M} \widehat{P}_{i_0j} \frac{|\psi[j]|}{|\psi[i_0]|} \leq \sum_{j=1}^{2M} \widehat{P}_{i_0j} = 1$. The first inequality is due to the triangle inequality and the second equality is due to the kernel normalization by $\widehat{\mathbf{D}}^{-1}$. □

Theorem 5 shows that the eigenvalues are bounded. However, bounded eigenvalues are insufficient for dimensionality reduction. Dimensionality reduction is meaningful when there is a significant spectral decay.

Defenition 1. *Let \mathcal{M} be a manifold. The intrinsic dimension d of the manifold is a positive integer determined by how many independent “coordinates” are needed to describe \mathcal{M} . Using a parametrization to describe a manifold, the dimension of \mathcal{M} is the smallest integer d such that a smooth map $\mathbf{f}(\xi) = \mathcal{M}$, $\xi \in \mathcal{R}^d$, describes the manifold, where $\xi \in \mathcal{R}^d$.*

Our framework is based on a Gaussian kernel. The spectral decay of Gaussian kernels was studied in [5]. We use Lemma 2 to evaluate the spectral decay of our kernel.

Lema 2. *Assume that the data is sampled from a manifold with intrinsic dimension $d \ll M$. Let \mathbf{K}° (section III-B) denotes the kernel with an exponential decay as a function of the Euclidean distance. For $\delta > 0$, the number of eigenvalues of \mathbf{K}° above δ is proportional to $(\log(\frac{1}{\delta}))^d$.*

Lemma 2 is based on Weyl's asymptotic law [30]. Let $r_\delta = r(\delta) = \max\{\ell \in N \text{ such that } |\lambda_\ell| \geq \delta\}$ denotes the number of eigenvalues of \mathbf{K}° above δ . $K_{i,j}^\circ = K_{i,j}^x K_{i,j}^y$ corresponds to a single DM view given in [5]. Theorem 6 relates the spectral decay of the kernel $\widehat{\mathbf{P}}$ from (Eq. (6)) to the decay of the Kernel Product-based DM (\mathbf{P}° Eq. (7) and in [5]).

Lema 3. *Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{M \times M}$ be such that $\mathbf{A}, \mathbf{B} \geq 0$. Then for any $1 \leq k \leq M-1$*

$$\prod_{\ell=k}^{M-1} \lambda_\ell(\mathbf{A} \cdot \mathbf{B}) \leq \prod_{\ell=k}^{M-1} \lambda_\ell(\mathbf{A} \circ \mathbf{B}) \quad (21)$$

where \circ is the Kronecker matrix product.

This inequality is proved in [34] and [35].

Theorem 6. *Multiplying the last $M - 1 - r_\delta$ eigenvalues of \mathbf{K}^z is smaller than δ^{M-1-r_δ} . Formally,*

$$\prod_{\ell=r_\delta}^{M-1} \lambda_\ell(\mathbf{K}^x \cdot \mathbf{K}^y) \leq \delta^{M-1-r_\delta}.$$

Proof. Denote by $\{\lambda_i(\mathbf{A})\}_{i=0}^{M-1}$ the eigenvalues of the matrix \mathbf{A} . They are enumerated in descending order such that $\lambda_0(\mathbf{A}) \geq \lambda_2(\mathbf{A}) \geq \dots \geq \lambda_{M-1}(\mathbf{A})$. We use Lemma 3 to prove Theorem 6 by choosing $\mathbf{A} = \mathbf{K}^x$ and $\mathbf{B} = \mathbf{K}^y$, which are positive semi-definite. $\mathbf{K}^x \circ \mathbf{K}^y = \mathbf{K}^\circ$ corresponds to the approach in [5]. By using Lemma 3 and choosing $\ell = r_\delta$ in Eq. 21, we get

$$\prod_{\ell=r_\delta}^{M-1} \lambda_\ell(\mathbf{K}^x \cdot \mathbf{K}^y) \leq \prod_{\ell=r_\delta}^{M-1} \lambda_\ell(\mathbf{K}^\circ) \leq \delta^{M-1-r_\delta}.$$

□

Using the kernel matrix spectral decay, we can approximate Eq. (12) by neglecting all the eigenvalues that are smaller than δ . Thus, we can compute a low dimensional mapping such that

$$\widehat{\Psi}_t^r(\mathbf{x}_i) : \mathbf{x}_i \mapsto [\lambda_1^t \psi_1[i], \lambda_2^t \psi_2[i], \lambda_3^t \psi_3[i], \dots, \lambda_{r-1}^t \psi_{r-1}[i]]^T \in \mathbb{R}^{r-1}. \quad (22)$$

This mapping of dimension r provides a low dimensional space which improves the performance and the efficiency of various machine learning tasks. The following Lemma introduces an error bound for using low dimension mapping $\widehat{\Psi}_t^r(\mathbf{x}_i)$.

Lema 4. *The truncated diffusion distance up to coordinate r defined as*

$$[\mathcal{D}_t^r(\mathbf{x}_i, \mathbf{x}_j)]^2 \triangleq \|\widehat{\Psi}_t^r(\mathbf{x}_i) - \widehat{\Psi}_t^r(\mathbf{x}_j)\|^2 = \sum_{s=1}^r \lambda_s^{2t} (\psi_s[i] - \psi_s[j])^2, \quad (23)$$

is bounded by the inner view diffusion distance (defined in Eq. (10))

$$2 \cdot \left[\sum_{s=1}^{M-1} \lambda_s^{2t} (\psi_s[i] - \psi_s[j])^2 - \delta^{2t} \cdot \left(\frac{1 - \delta_{i,j}}{\widehat{D}} \right) \right] \leq [\mathcal{D}_t^r(\mathbf{x}_i, \mathbf{x}_j)]^2 \leq 2 \cdot \sum_{s=1}^{M-1} \lambda_s^{2t} (\psi_s[i] - \psi_s[j])^2,$$

where $\delta_{i,j}$ is the Kronecker delta function.

Proof. For the right inequality, clearly

$$[\mathcal{D}_t^r(\mathbf{x}_i, \mathbf{x}_j)]^2 \leq [\mathcal{D}_t^{2M}(\mathbf{x}_i, \mathbf{x}_j)]^2 = 2 \cdot \sum_{s=1}^{M-1} \lambda_s^{2t} (\psi_s[i] - \psi_s[j])^2,$$

the equality is a result of the repetitive form of Ψ which was defined in Theorem 3 for $L = 2$. Note that $s = 0$ was excluded from the sum as $\psi_0 = 1$ is constant. For the left inequality, using that $\Psi = \widehat{D}^{-1/2} \Pi$, where Π is an orthonormal basis defined in Eq. (19) by using the orthogonality of Π we get that

$$\Psi \Psi^T = \widehat{D}^{-1/2} \Pi \Pi^T \widehat{D}^{-1/2} = \widehat{D}^{-1},$$

which means that

$$\sum_{s=0}^{2M-1} (\psi_s[i] - \psi_s[j])^2 = \frac{1}{\widehat{D}_{i,i}} + \frac{1}{\widehat{D}_{j,j}} - \frac{2\delta_{i,j}}{\widehat{D}_{i,i}}. \quad (24)$$

By the definition of the truncated diffusion distance we have

$$[\mathcal{D}_t^r(\mathbf{x}_i, \mathbf{x}_j)]^2 = \sum_{s=0}^{2M-1} \lambda_s^{2t} (\psi_s[i] - \psi_s[j])^2 - \sum_{s=r+1}^{2M-1} \lambda_s^{2t} (\psi_s[i] - \psi_s[j])^2 \geq$$

$$2 \cdot \sum_{s=1}^{M-1} \lambda_s^{2t} (\psi_s[i] - \psi_s[j])^2 - \delta^{2t} \sum_{s=0}^{2M-1} (\psi_s[i] - \psi_s[j])^2 \geq 2 \cdot \left[\sum_{s=1}^{2M-1} \lambda_s^{2t} (\psi_s[i] - \psi_s[j])^2 - \delta^{2t} \cdot \left(\frac{1 - \delta_{i,j}}{\tilde{D}} \right) \right],$$

where \tilde{D} is the minimal value of $\hat{D}_{i,i}$ and $\hat{D}_{j,j}$. In the same way, a bound for the truncated diffusion distance between \mathbf{y}_i and \mathbf{y}_j is derived. \square

E. Out-of-sample extension

To extend the diffusion coordinates to new data points without re-applying a large-scale eigendecomposition [31], the Nyström extension is widely used. Here we formulate the extension method for a multi-view scenario. Given the data sets \mathbf{X} and \mathbf{Y} and new points $\tilde{\mathbf{x}} \notin \mathbf{X}$ and $\tilde{\mathbf{y}} \notin \mathbf{Y}$, we want to extend the multi-view diffusion mapping to $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ without re-applying the proposed framework. First, we describe the explicit form for the eigenvalue problem for two views \mathbf{X} and \mathbf{Y} . The eigenvector ψ_k with the corresponding eigenvalue λ_k satisfies $\lambda_k \psi_k = \hat{\mathbf{P}} \psi_k$. By using the definition of $\hat{\mathbf{P}}$ from Eqs. (4) and (6) we get that

$$\lambda_k \psi_k = \hat{\mathbf{D}}^{-1} \begin{bmatrix} \mathbf{0}_{M \times M} & \mathbf{K}^x \cdot \mathbf{K}^y \\ \mathbf{K}^y \cdot \mathbf{K}^x & \mathbf{0}_{M \times M} \end{bmatrix} \cdot \psi_k,$$

due to the block form of the matrix $\hat{\mathbf{P}}$

$$\lambda_k \psi_k^x[i] = \sum_j \hat{p}(\mathbf{x}_i, \mathbf{y}_j) \psi_k^y[j],$$

$$\lambda_k \psi_k^y[i] = \sum_j \hat{p}(\mathbf{y}_i, \mathbf{x}_j) \psi_k^x[j],$$

where $\psi_k^x[i] \triangleq \psi_k[i]$, $i = 1, \dots, M$ and $\psi_k^y[i] \triangleq \psi_k[i + M]$, $i = 1, \dots, M$. The transition matrices are

$$\hat{p}(\mathbf{x}_i, \mathbf{y}_j) = \frac{\sum_s K_{i,s}^x K_{s,j}^y}{\hat{D}_{i,i}^{rows}}, \text{ and } \hat{p}(\mathbf{y}_i, \mathbf{x}_j) = \frac{\sum_s K_{i,s}^y K_{s,j}^x}{\hat{D}_{i,i}^{cols}}.$$

The Nyström extension is an approximated weighted sum of the original eigenvectors. The weights are computed by applying the kernel $\hat{\mathcal{P}}$ to the extended data points. For the proposed mapping, the extension is defined by

$$\hat{\psi}_k(\tilde{\mathbf{x}}) = \frac{1}{\lambda_k} \sum_j \hat{\mathcal{P}}(\tilde{\mathbf{x}}, \mathbf{y}_j) \psi_k^y[j] = \frac{1}{\lambda_k} \sum_j \sum_s \exp \left\{ -\frac{\|\tilde{\mathbf{x}} - \mathbf{x}_s\|^2}{2\sigma_x^2} \right\} \exp \left\{ -\frac{\|\mathbf{y}_s - \mathbf{y}_j\|^2}{2\sigma_y^2} \right\} \frac{\psi_k[j + M]}{\tilde{D}_j} \quad (25)$$

$$\hat{\psi}_k(\tilde{\mathbf{y}}) = \frac{1}{\lambda_k} \sum_j \hat{\mathcal{P}}(\tilde{\mathbf{y}}, \mathbf{x}_j) \psi_k^x[j] = \frac{1}{\lambda_k} \sum_j \sum_s \exp \left\{ -\frac{\|\tilde{\mathbf{y}} - \mathbf{y}_s\|^2}{2\sigma_y^2} \right\} \exp \left\{ -\frac{\|\mathbf{x}_s - \mathbf{x}_j\|^2}{2\sigma_x^2} \right\} \frac{\psi_k[j]}{\tilde{D}_j} \quad (26)$$

and the new mapping vector for data point is

$$\hat{\Psi}(\tilde{\mathbf{x}}) = [\lambda_1 \hat{\psi}_1(\tilde{\mathbf{x}}), \lambda_2 \hat{\psi}_2(\tilde{\mathbf{x}}), \lambda_3 \hat{\psi}_3(\tilde{\mathbf{x}}), \dots, \lambda_{M-1} \hat{\psi}_{M-1}(\tilde{\mathbf{x}})] \in \mathbb{R}^{M-1}. \quad (27)$$

The new coordinates in the diffusion space are approximated and the new data points $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ have no effect on the original map's structure.

F. Infinitesimal generator

A family of diffusion operators were introduced in [5]. Each operator differs by the normalization applied to it. If appropriate limits are taken such that $M \rightarrow \infty$, $\epsilon \rightarrow 0$ and $\epsilon = 2\sigma^2$, then from [5] it follows that the DM kernel operator will converge to one of the following differential operators: 1. Normalized graph Laplacian. 2. Laplace-Beltrami diffusion. 3. Heat kernel equation. These are proved in [5]. The operators are all special cases of the diffusion equation. This convergence provides not only a physical justification for the DM framework, but allows in some cases to distinguish between the geometry and the density of the data points. In this section, we study the asymptotic properties of the proposed kernel \widehat{K} (Eq. (5)) by using only two views, i.e. $L = 2$.

We are interested in understanding the properties of the eigenfunctions of the proposed multi-view kernel \widehat{P} (Eqs. (5), (6)) for two views. We assume that there is some unknown mapping $\beta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ from view \mathbf{X} to view \mathbf{Y} that satisfies $\mathbf{y}_i = \beta(\mathbf{x}_i)$, $i = 1, \dots, M$. Each view-specific kernel function has the same properties $K^x = K^y = K$ such that $K \geq 0$, $K(\mathbf{z}) = K(-\mathbf{z})$ and the kernel is normalized such that $\int_{\mathbb{R}^d} K(\mathbf{z}) d\mathbf{z} = 1$. Note that with proper normalizations the Gaussian kernel satisfies these requirements. The analysis is performed for data points $\{\mathbf{x}_1, \dots, \mathbf{x}_M\} \in \mathbb{R}^D$ sampled from a uniform distribution over a bounded domain in \mathbb{R}^d . The image of the function β is a bounded domain in \mathbb{R}^d with distribution $\alpha(\mathbf{z})$.

Theorem 7. *The infinitesimal generator induced by the proposed kernel \widehat{P} (Eq. (5)) converges when $M \rightarrow \infty$, $\epsilon \rightarrow 0$, $\epsilon = 2\sigma_x^2 = 2\sigma_y^2$ to a “cross domain Laplacian operator”. The convergence is to functions $f(\mathbf{x})$ and $g(\mathbf{y})$, which are the eigenfunctions of \widehat{P} . These functions are the solutions of the following diffusion like equations:*

$$(\widehat{P}f)(\mathbf{x}_i) = g(\beta(\mathbf{x}_i)) + \epsilon \Delta \gamma(\beta(\mathbf{x}_i)) / \alpha(\beta(\mathbf{x}_i)) + \mathcal{O}(\epsilon^{3/2}), \quad (28)$$

$$(\widehat{P}g)(\mathbf{y}_i) = f(\beta^{-1}(\mathbf{y}_i)) + \epsilon \Delta \eta(\beta^{-1}(\mathbf{y}_i)) / \alpha(\beta(\mathbf{y}_i)) + \mathcal{O}(\epsilon^{3/2}), \quad (29)$$

where the functions γ, η are defined $\gamma(\mathbf{z}) \triangleq g(\mathbf{z})\alpha(\mathbf{z})$, $\eta(\mathbf{z}) \triangleq f(\mathbf{z})\alpha(\mathbf{z})$.

Proof. The eigenfunction of the operator \widehat{L} is defined using the functions $f(\mathbf{x})$ and $g(\mathbf{y})$ by concatenating the vectors such that

$$\mathbf{h} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_M), g(\mathbf{y}_1), g(\mathbf{y}_2), \dots, g(\mathbf{y}_M)] \in \mathbb{R}^{2M}.$$

By expanding the single view construction presented in [5], [33]. The limit of the characteristic equation is

$$\lim_{\substack{M \rightarrow \infty \\ \epsilon \rightarrow 0}} (\widehat{L}h_i) = \lim_{\substack{M \rightarrow \infty \\ \epsilon \rightarrow 0}} h_i - \frac{\sum_{j=1}^{2M} \widehat{K}_{i,j} h_j}{\sum_{j=1}^{2M} \widehat{K}_{i,j}} = \lim_{\substack{M \rightarrow \infty \\ \epsilon \rightarrow 0}} f(\mathbf{x}_i) - \frac{\sum_{j=1}^M \sum_{\ell=1}^M K_{i,\ell}^x K_{\ell,j}^y g(\mathbf{y}_j)}{\sum_{j=1}^M \sum_{\ell=1}^M K_{i,\ell}^x K_{\ell,j}^y}, i = 1, \dots, M. \quad (30)$$

We approximate the summation based on a Riemann integral. Thus, the denominator becomes

$$\frac{1}{M^2 \epsilon^d} \sum_{j=1}^M \sum_{\ell=1}^M K_{i,\ell}^x K_{\ell,j}^y \xrightarrow[\epsilon \rightarrow 0]{M \rightarrow \infty} \frac{1}{\epsilon^d} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} K\left(\frac{\mathbf{s} - \mathbf{x}}{\sqrt{\epsilon}}\right) K\left(\frac{\mathbf{y} - \beta(\mathbf{s})}{\sqrt{\epsilon}}\right) \alpha(\mathbf{y}) d\mathbf{s} d\mathbf{y}.$$

Using a change of variables $\mathbf{z} = \frac{\mathbf{y} - \beta(\mathbf{s})}{\sqrt{\epsilon}}$, $\mathbf{y} = \beta(\mathbf{s}) + \sqrt{\epsilon}\mathbf{z}$, $d\mathbf{z} = d\mathbf{y}\epsilon^{d/2}$ we get

$$\frac{1}{\epsilon^d} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} K\left(\frac{\mathbf{s} - \mathbf{x}}{\sqrt{\epsilon}}\right) K\left(\frac{\mathbf{y} - \beta(\mathbf{s})}{\sqrt{\epsilon}}\right) \alpha(\mathbf{y}) d\mathbf{s} d\mathbf{y} = \frac{1}{\epsilon^{d/2}} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} K\left(\frac{\mathbf{s} - \mathbf{x}}{\sqrt{\epsilon}}\right) K(\mathbf{z}) \alpha(\beta(\mathbf{s}) + \sqrt{\epsilon}\mathbf{z}) d\mathbf{s} d\mathbf{z}.$$

Using a first order Taylor expansion of α we get

$$\approx \frac{1}{\epsilon^{d/2}} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} K\left(\frac{\mathbf{s} - \mathbf{x}}{\sqrt{\epsilon}}\right) K(\mathbf{z}) [\alpha(\beta(\mathbf{s})) + \frac{\sqrt{\epsilon}}{2} \mathbf{z}^T \nabla \alpha(\beta(\mathbf{s})) + \mathcal{O}(\epsilon)] d\mathbf{s} d\mathbf{z},$$

using the symmetry of the kernel $K(\mathbf{z})$ we get

$$\frac{\sqrt{\epsilon}}{2} \int_{\mathbb{R}^d} K(\mathbf{z}) \mathbf{z}^T \nabla \alpha(\beta(\mathbf{s})) d\mathbf{z} = 0.$$

Applying the change of variables $\mathbf{t} = \frac{\mathbf{s} - \mathbf{x}}{\sqrt{\epsilon}}$, $\mathbf{s} = \sqrt{\epsilon} \mathbf{t} + \mathbf{x}$, $d\mathbf{t} = d\mathbf{s} \epsilon^{d/2}$ we get

$$\approx \int_{\mathbb{R}^d} K(\mathbf{t}) [\alpha(\beta(\mathbf{x} + \sqrt{\epsilon} \mathbf{t})) + \mathcal{O}(\epsilon)] d\mathbf{t} \approx \alpha(\beta(\mathbf{x})) + \mathcal{O}(\epsilon).$$

The last transition is based on a Taylor expansion and zeroing out odd moments of $K(\mathbf{t})$. The Riemann integral for the nominator from Eq. (30) takes the following form

$$\frac{1}{M^2 \epsilon^d} \sum_{j=1}^M \sum_{\ell=1}^M K_{i,\ell}^x K_{\ell,j}^y g(\mathbf{y}_j) \xrightarrow[\epsilon \rightarrow 0]{M \rightarrow \infty} \frac{1}{\epsilon^d} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} K\left(\frac{\mathbf{s} - \mathbf{x}}{\sqrt{\epsilon}}\right) K\left(\frac{\mathbf{y} - \beta(\mathbf{s})}{\sqrt{\epsilon}}\right) g(\mathbf{y}) \alpha(\mathbf{y}) d\mathbf{s} d\mathbf{y}.$$

By applying a change of variables

$\mathbf{z} = \frac{\mathbf{y} - \beta(\mathbf{s})}{\sqrt{\epsilon}}$, $\mathbf{y} = \beta(\mathbf{s}) + \sqrt{\epsilon} \mathbf{z}$, $d\mathbf{z} = d\mathbf{y} \epsilon^{d/2}$ we get

$$\frac{1}{\epsilon^{d/2}} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} K\left(\frac{\mathbf{s} - \mathbf{x}}{\sqrt{\epsilon}}\right) K(\mathbf{z}) \gamma(\beta(\mathbf{s}) + \sqrt{\epsilon} \mathbf{z}) d\mathbf{s} d\mathbf{z}.$$

By using Taylor's expansion of $\gamma(\beta(\mathbf{s}))$ we get

$$\underbrace{\approx \frac{1}{\epsilon^{d/2}} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} K\left(\frac{\mathbf{s} - \mathbf{x}}{\sqrt{\epsilon}}\right) K(\mathbf{z}) [\gamma(\beta(\mathbf{s})) + \frac{\sqrt{\epsilon}}{2} \mathbf{z}^T \nabla \gamma(\beta(\mathbf{s})) + \frac{\epsilon}{2} \mathbf{z}^T \mathbf{H} \mathbf{z} + \mathcal{O}(\epsilon^{3/2})] d\mathbf{s} d\mathbf{z}}_{\text{L}}$$

where $H_{i,j} \triangleq \frac{\partial^2 \gamma(\beta(\mathbf{s}))}{\partial s_i \partial s_j}$ is the Hessian. The first term is the integral over $K(\mathbf{z})$, while the second

$$\frac{\sqrt{\epsilon}}{2} \int_{\mathbb{R}^d} K(\mathbf{z}) \mathbf{z}^T \nabla \gamma(\beta(\mathbf{s})) d\mathbf{z} = 0$$

due to the symmetry of the kernel $K(\mathbf{z})$. The last term becomes

$$\int_{\mathbb{R}^d} K(\mathbf{z}) \mathbf{z}^T \frac{\partial \gamma(\beta(\mathbf{s}))}{\partial s_i \partial s_j} \mathbf{z} d\mathbf{z} = \sum_{i,j} \frac{\partial \gamma(\beta(\mathbf{s}))}{\partial s_i \partial s_j} \int_{\mathbb{R}^d} z_i z_j K(\mathbf{z}) d\mathbf{z} = \sum_i \frac{\partial^2 \gamma(\beta(\mathbf{s}))}{\partial s_i^2} \int_{\mathbb{R}^d} z_i^2 K(\mathbf{z}) d\mathbf{z} = \Delta \gamma(\beta(\mathbf{s})).$$

We substitute the results in (L) to get

$$\frac{1}{\epsilon^{d/2}} \int_{\mathbb{R}^d} K\left(\frac{\mathbf{s} - \mathbf{x}}{\sqrt{\epsilon}}\right) [\gamma(\beta(\mathbf{s})) + \frac{\epsilon}{2} \Delta \gamma(\beta(\mathbf{s})) + \mathcal{O}(\epsilon^{3/2})] d\mathbf{s}.$$

By applying a change of variables $\mathbf{t} = \frac{\mathbf{s} - \mathbf{x}}{\sqrt{\epsilon}}$, $\mathbf{s} = \sqrt{\epsilon} \mathbf{t} + \mathbf{x}$, $d\mathbf{t} = d\mathbf{s} \epsilon^{d/2}$ and $\gamma(\mathbf{y}) = g(\mathbf{y}) \alpha(\mathbf{y})$ we get

$$\int_{\mathbb{R}^d} K(\mathbf{t}) [\gamma(\beta(\mathbf{x} + \sqrt{\epsilon} \mathbf{t})) + \frac{\epsilon}{2} \Delta \gamma(\beta(\mathbf{x} + \sqrt{\epsilon} \mathbf{t})) + \mathcal{O}(\epsilon^{3/2})] d\mathbf{t}.$$

By using Taylor's expansion again we get

$$\approx \int_{\mathbb{R}^d} K(\mathbf{t}) [\gamma(\beta(\mathbf{x})) + \frac{\epsilon}{2} \Delta \gamma(\beta(\mathbf{x})) + \frac{\epsilon}{2} \mathbf{t}^T \mathbf{H} \mathbf{t} + \mathcal{O}(\epsilon^{3/2})] d\mathbf{t}.$$

We neglected the order terms when ϵ is raised to a power higher than 3/2. Terms with odd order of \mathbf{t} are zeroed due to the symmetry of the kernel K . Using the same argument as in the integral (L) we get that the nominator is

$$\gamma(\beta(\mathbf{x})) + \epsilon \Delta \gamma(\beta(\mathbf{x})) + \mathcal{O}(\epsilon^{3/2}),$$

dividing by the denominator we get

$$(\hat{P}f)(\mathbf{x}_i) = g(\beta(\mathbf{x}_i)) + \epsilon \Delta \gamma(\beta(\mathbf{x}_i)) / \alpha(\beta(\mathbf{x}_i)) + \mathcal{O}(\epsilon^{3/2})$$

In the same way, we compute the convergence on $g(\mathbf{y}_i)$

$$(\hat{P}g)(\mathbf{y}_i) = f(\beta^{-1}(\mathbf{y}_i)) + \epsilon \Delta \eta(\beta^{-1}(\mathbf{y}_i)) / \alpha(\beta(\mathbf{y}_i)) + \mathcal{O}(\epsilon^{3/2}).$$

□

We ignored in the above computation the following

- Error due to approximating the sum by an integral, a bound for such error in the single view DM is introduced in [32].
- Deformation due to the fact that the data is sampled from a non uniform density. This changes the result by some constant.
- The data lies on some manifold. This could be dealt by changing the coordinate system and integrating on the manifold.
- When assuming that the data lies on some manifold, the Euclidean distance should be replaced by the geodesic distance along the manifold. As in the analysis of [5], this introduces a factor to the integral.

G. The convergence rate

In Theorem 7, we assume the number of data points $M \rightarrow \infty$ while the scale parameter $\epsilon \rightarrow 0$. In practice we cannot expect to have an infinite number of data points. It was shown in [5], [36] and others that a single view graph Laplacian converges to the laplacian operator on a manifold. It is demonstrated in [37], [38] that the variance of the error for such operator decreases as $M \rightarrow \infty$, but increases as $\epsilon \rightarrow 0$. The study in [37] proves that for a uniform distribution of data points, the variance of the error is bounded by $\mathcal{O}(\frac{1}{M^{1/2}\epsilon^{1+d/4}}, \epsilon^{1/2})$. This bound was improved in [38] by an asymptotic factor of $\sqrt{\epsilon}$ based on the correlation between \mathbf{D}^{-1} and \mathbf{K} .

We now turn our attention to the variance of the multi-view kernel for a finite number of points. Given $\mathbf{x}_1, \dots, \mathbf{x}_M$ independent uniformly distributed data points sampled from a bounded domain in \mathbb{R}^d . Define the multi-view Parzen Window density estimator by

$$\dot{\mathbf{K}}_{M,\epsilon}(\mathbf{x}) \triangleq \frac{1}{M^2} \sum_{\ell=1}^M \sum_{j=1}^M \frac{1}{\epsilon^d} K\left(\frac{\mathbf{x} - \mathbf{x}_\ell}{\sqrt{\epsilon}}\right) K\left(\frac{\mathbf{y}_\ell - \mathbf{y}_j}{\sqrt{\epsilon}}\right). \quad (31)$$

We are interested in finding a bound for the variance of $\dot{\mathbf{K}}_{M,\epsilon}(\mathbf{x})$ for a finite value of data points.

$$\begin{aligned} \text{Var}(\dot{\mathbf{K}}_{M,\epsilon}(\mathbf{x})) &= \frac{1}{M^4 \epsilon^{2d}} \cdot M \cdot \text{Var} \left[\sum_{\ell}^M K\left(\frac{\mathbf{x} - \mathbf{x}_\ell}{\sqrt{\epsilon}}\right) K\left(\frac{\mathbf{y}_\ell - \mathbf{y}_j}{\sqrt{\epsilon}}\right) \right] \leq \\ &\frac{1}{M^4 \epsilon^{2d}} \cdot M^3 \cdot \text{Var} [K_\epsilon^x K_\epsilon^y] \leq \frac{1}{M \epsilon^{2d}} [\text{Var}(K_\epsilon^x) \cdot \|K_\epsilon^y\|_\infty + \text{Var}(K_\epsilon^y) \cdot \|K_\epsilon^x\|_\infty] \leq \\ &\frac{1}{M \epsilon^{2d}} \cdot [\epsilon^{d/2} \cdot m_1 \cdot 1 + \epsilon^{d/2} \cdot m_2 \cdot \alpha(\mathbf{x})] \leq \frac{m_1 + m_2 \cdot \alpha(\mathbf{x})}{M \cdot \epsilon^{1.5d}}. \end{aligned}$$

The constants m_1 and m_2 are functions of the kernels choice, and the function α of the density of points $\mathbf{y}_i, i = 1, \dots, M$. This bound helps to choose an optimal value for the scaling factor ϵ given the number of data points M and the intrinsic dimension d .

H. Generalized multi-view kernel

One can consider a more general multi-view kernel such that it enables a transition within views \mathbf{X} and \mathbf{Y} in each time step. Such a kernel will take the following form

$$\widehat{\mathbf{K}} = \begin{bmatrix} (1 - \alpha) \cdot (\mathbf{K}^x)^2 & \alpha \cdot \mathbf{K}^x \mathbf{K}^y \\ \alpha \cdot \mathbf{K}^y \mathbf{K}^x & (1 - \alpha) \cdot (\mathbf{K}^y)^2 \end{bmatrix}, \quad (32)$$

where the parameter $\alpha \in [0, 1]$ provides a bias for the within view transition probability. This kernel is normalized using the sum of rows diagonal matrix $\widehat{\mathbf{D}}$, such that $\widehat{\mathbf{P}} = \widehat{\mathbf{D}}^{-1} \widehat{\mathbf{K}}$. For a small value of α , the kernel favors the within view transition probability. Therefore share similar properties with the single view diffusion process. For a large value of α , the kernel behaves empirically like the multi-view kernel $\widehat{\mathbf{K}}$ (Eq. (5)).

V. RELATED WORK

In this section, we describe alternative frameworks that incorporate multiple views. The first two are related to the diffusion process, however, the proposed kernels are not symmetric thus they do not guarantee real eigenvalues and eigenvectors.

A. Unsupervised metric fusion by cross diffusion

The framework in [13] suggests to construct two matrices for each view $\mathbf{P}^1, \mathbf{P}^2$, and $\mathcal{P}^1, \mathcal{P}^2$. The first pair is defined as in Eq. (1). The second pair is a stochastic matrix computed using only K^{NN} nearest neighbors for each instance. The kernels are fused such that

$$\mathbf{P}_{t+1}^1 = \mathcal{P}^1 \cdot \mathbf{P}_t^2 \cdot (\mathcal{P}^1)^T, \quad (33)$$

$$\mathbf{P}_{t+1}^2 = \mathcal{P}^2 \cdot \mathbf{P}_t^1 \cdot (\mathcal{P}^2)^T. \quad (34)$$

The diffusion process incorporates two steps (between the view) in each time unit. A convergence of the induced diffusion distance is proved for $t \rightarrow \infty$ in [13].

B. Common manifold learning using alternating-diffusion

An alternating diffusion process is proposed in [39]. The construction is based on fusing the stochastic matrices \mathbf{P}^1 and \mathbf{P}^2 by multiplying the matrices such that $\mathbf{P}^{AD} = \mathbf{P}^1 \cdot \mathbf{P}^2$. Assuming that a common random variable exists in both views, new results regarding the extraction of underlying hidden random parameters are described in [39]. The study in [39] was inspired by our work by giving a reference to the current paper.

C. Kernel Canonical Correlation Analysis (KCCA)

The frameworks [40], [41] extend the well know Canonical Correlation Analysis (CCA) by applying a kernel function prior to the application of CCA. Kernels \mathbf{K}^1 and \mathbf{K}^2 are constructed for each view as in Eq. (4) and the canonical vectors \mathbf{v}_1 and \mathbf{v}_2 are computed by solving the following generalized eigenvalue problem

$$\begin{bmatrix} \mathbf{0}_{M \times M} & \mathbf{K}^1 \cdot \mathbf{K}^2 \\ \mathbf{K}^2 \cdot \mathbf{K}^1 & \mathbf{0}_{M \times M} \end{bmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} = \rho \cdot \begin{bmatrix} (\mathbf{K}^1 + \gamma \mathbf{I})^2 & \mathbf{0}_{M \times M} \\ \mathbf{0}_{M \times M} & (\mathbf{K}^2 + \gamma \mathbf{I})^2 \end{bmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix}, \quad (35)$$

where $\gamma \mathbf{I}$ are regularization terms which guarantee that the matrices $(\mathbf{K}^1 + \gamma \mathbf{I})^2$ and $(\mathbf{K}^2 + \gamma \mathbf{I})^2$ are invertible. Usually the Incomplete Cholesky Decomposition (ICD) [40], [41], [42] is used to reduce the run time required for solving (35). For clustering tasks, K- means is applied to the set of generalized eigenvectors.

D. Spectral clustering with two views

The approach in [18] generalizes the traditional normalized graph Laplacian for two views. Kernels \mathbf{K}^1 and \mathbf{K}^2 are computed in each view as in Eq. (4). The kernels are multiplied such that $\mathbf{W} = \mathbf{K}^1 \cdot \mathbf{K}^2$ and

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{M \times M} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{0}_{M \times M} \end{bmatrix}. \quad (36)$$

Finally, normalization is done by using $\bar{\mathbf{D}}$ where $\bar{D}_{i,i} = \sum_j W_{i,j}$, such that the normalized fused kernel is defined as

$$\bar{\mathbf{A}} = \bar{\mathbf{D}}^{-0.5} \cdot \mathbf{A} \cdot \bar{\mathbf{D}}^{-0.5}. \quad (37)$$

Assuming that the number of clusters in the data is N_C , denoting the eigenvectors of \mathbf{A} as $\phi_i, i = 1, \dots, 2M$. K-Means algorithm is applied to the mapping

$$\Phi[i] \triangleq [\phi_1^2[i], \dots, \phi_{N_C}^2[i]]/s[i], \quad (38)$$

where $s[i] = \sum_j^{N_C} (\phi_j^2[i])$. The paper in [18] is focused on spectral clustering, however, a similar version of the kernel from Eq. (37) is suited for manifold learning as we demonstrate in this study. We use a stochastic version of the kernel and extend the construction to multiple views. The stochastic version $\hat{\mathbf{P}}$ (Eq. (6)) is useful as it provides various theoretical justifications for the multi-view diffusion process. In section VI this approach is referred as De Sa's.

VI. EXPERIMENTAL RESULTS

In this section, we present the experimental results which evaluate our framework. We focus the experiments on three tasks in machine learning: clustering, classification and manifold learning.

A. Empirical evaluations of theoretical aspects

In the first group of experiments we provide empirical evidence which support the theoretical analysis from Section IV.

1) *Spectral decay*: In Section IV-D, an upper bound on the eigenvalues' decay rate for our multi-view-based approach (matrix $\hat{\mathbf{P}}$ Eq. (6)) is presented. In order to empirically evaluate the decay rate, synthetic datasets are generated accompanied by comparison to other approaches. To evaluate the spectral decay of $\hat{\mathbf{P}}$ (Eq. (6)), \mathbf{P}° (Eq. (7) and [5]) and \mathbf{P}^+ (Eq. (9)) we compare the spectral decay rate between various frameworks on synthetic clustered data drawn from Gaussian distributions. The following steps describe the generation of both views denoted by (\mathbf{X}, \mathbf{Y}) and referred as View-I (\mathbf{X}) and View-II (\mathbf{Y}), respectively:

- 1) 6 vectors $\mu_j \in \mathbb{R}^9, j = 1, \dots, 6$ were drawn from a Gaussian distribution $N(\mathbf{0}, 8 \cdot \mathbf{I}_{9 \times 9})$. These vectors are the center of masses of the generated classes.
- 2) 100 data points were drawn for each cluster j by using $\mu_j, 1 \leq j \leq 6$, from a Gaussian distribution $N(\mu_j, 2 \cdot \mathbf{I}_{9 \times 9})$. Denote these 600 data points by \mathbf{X} .
- 3) 100 data points were drawn for each cluster j by using $\mu_j, 1 \leq j \leq 6$, from a Gaussian distribution $N(\mu_j, 2 \cdot \mathbf{I}_{9 \times 9})$. Denote these 600 data points by \mathbf{Y} .

The first 3 dimensions of both views are presented in Fig. 4. We compute the probability matrix for each view \mathbf{P}^x and \mathbf{P}^y (Eq. (1)), the Kernel Sum approach probability matrix \mathbf{P}^+ (Eq. (9)), the Kernel Product approach \mathbf{P}° (Eq. (7)) and the proposed approach $\hat{\mathbf{P}}$. The eigendecomposition is computed for all matrices. The resulting eigenvalues' decay rate are compared with the eigenvalues product from both views. To get a fair comparison between all the methods, we set the Gaussian scale parameter σ_x and σ_y in each view and then use these scales in all the methods. The vectors' variance in the concatenation

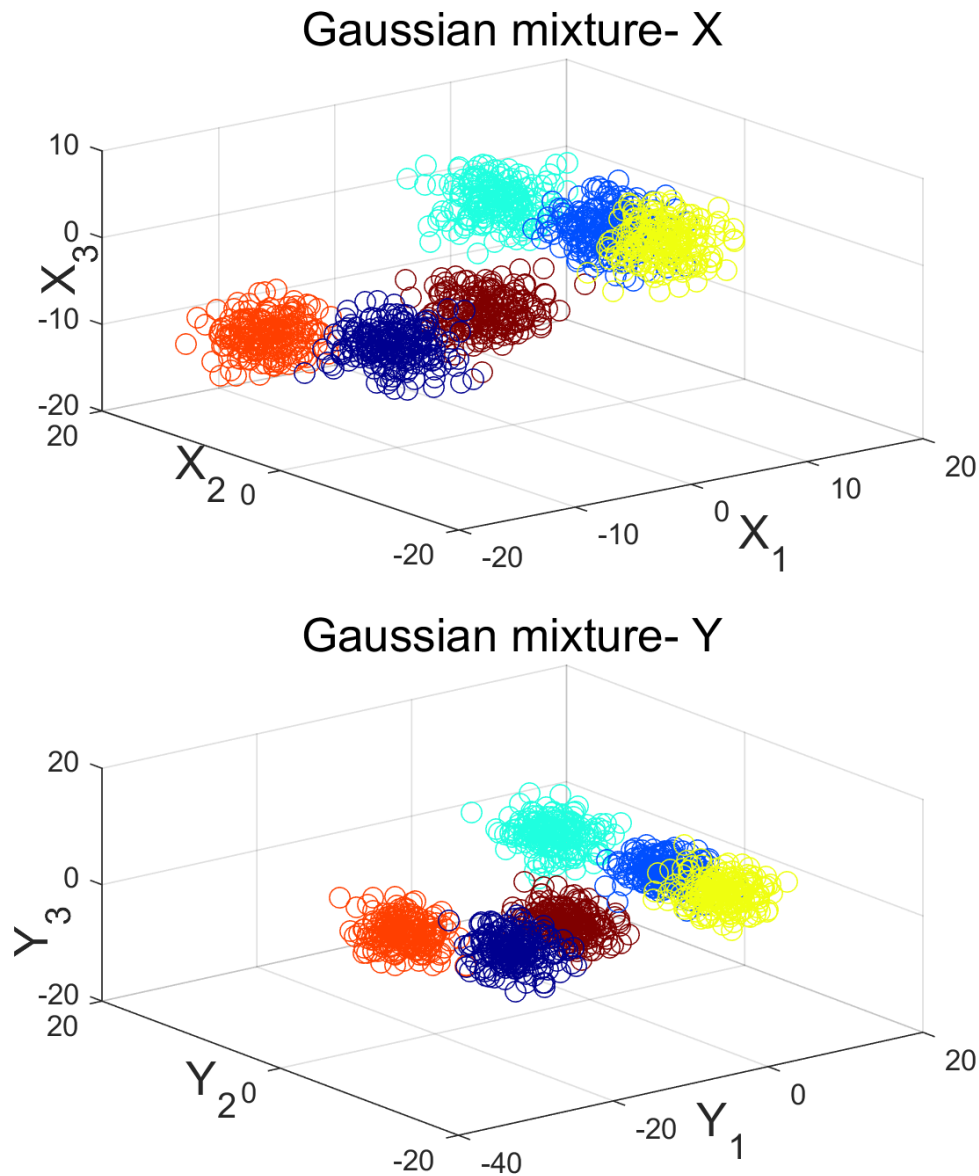


Fig. 4: The first 3 dimensions of the Gaussian mixture. Both views share the center of masses of the Gaussian spread. Top: first view denoted as \mathbf{X} . Bottom: second view denoted as \mathbf{Y} . The variance of the Gaussian in each dimension is 8.

approach is the sum of variances since we assume statistical independence. Therefore, the following scale parameters $\sigma_o^2 = \sigma_x^2 + \sigma_y^2$ are used.

The experiment is repeated but this time \mathbf{X} contains 6 clusters whereas \mathbf{Y} contains only 3. For \mathbf{Y} , we use only the first 3 center of masses and generate 200 points in each cluster. Figure 5 presents a logarithmic scale of the spectral decay for eigenvalues extracted from all methods. It is evident that our proposed kernel has the strongest spectral decay.

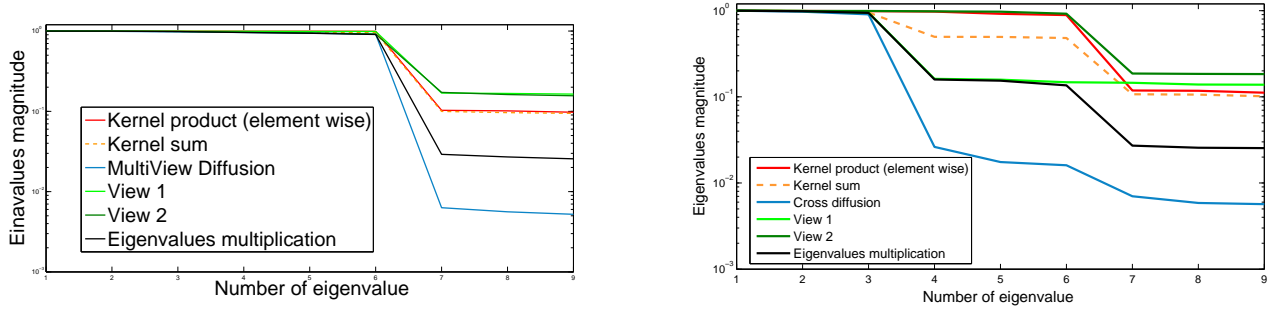


Fig. 5: Eigenvalues decay rate. Comparison between different mapping methods. Top: 6 clusters in each view. Bottom: 6 clusters in X and 3 clusters in Y .

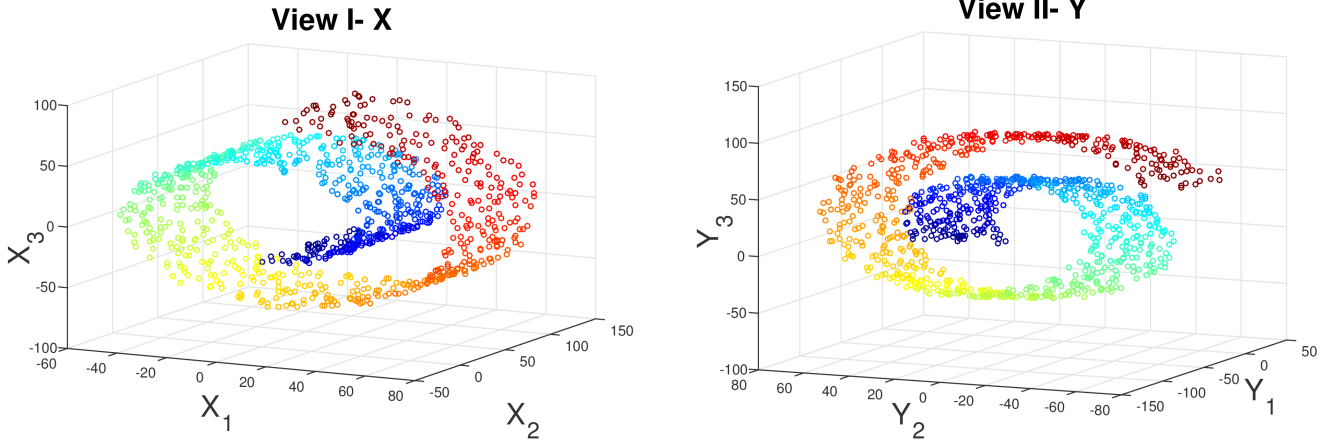


Fig. 6: The two Swiss Rolls, Top- generated by Eq. (39), Bottom- generated by Eq. (40).

2) *Cross view diffusion distance*: In this section, we examine the proposed Cross View Diffusion Distance (Section IV-C). A swiss roll is generated by using the function

$$\text{View I: } \mathbf{X} = \begin{bmatrix} x_i[1] \\ x_i[2] \\ x_i[3] \end{bmatrix} = \begin{bmatrix} 6\theta[i] \cos(\theta[i]) \\ h[i] \\ 6\theta[i] \sin(\theta[i]) \end{bmatrix} + \mathbf{N}_i^1, \quad (39)$$

$\theta_i = (1.5\pi)s_i$, $i = 1, 2, 3, \dots, 1000$, where s_i are 1000 data points that spread linearly within the line $s_i \rightarrow [1, 3]$. The second view is generated by the application of an orthonormal transformation to the swiss roll and adding Gaussian noise. The function in Eq. (40) describes the representation of the second view

$$\text{View II: } \mathbf{Y} = \begin{bmatrix} y_i[1] \\ y_i[2] \\ y_i[3] \end{bmatrix} = \mathbf{R} \begin{bmatrix} 6\theta_i \cos(\theta_i) \\ h_i \\ 6\theta_i \sin(\theta_i) \end{bmatrix} + \mathbf{N}_i^2, \quad (40)$$

where $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ is a random orthonormal transformation matrix. It is generated by drawing values from i.i.d Gaussian variables and applying the Graham-Schmidt process. $h_i, i = 1, \dots, 1000$ are drawn from a uniform distribution within the interval $[0, 100]$. Each component of $\mathbf{N}_i^1, \mathbf{N}_i^2 \in \mathbb{R}^{3 \times 1}$ is drawn from a Gaussian distribution with zero mean and a variance of σ_N^2 . An example for both Swiss rolls is presented in Fig. 6. A standard DM is applied to each view and a 2-dimensional embedding of the Swiss roll is

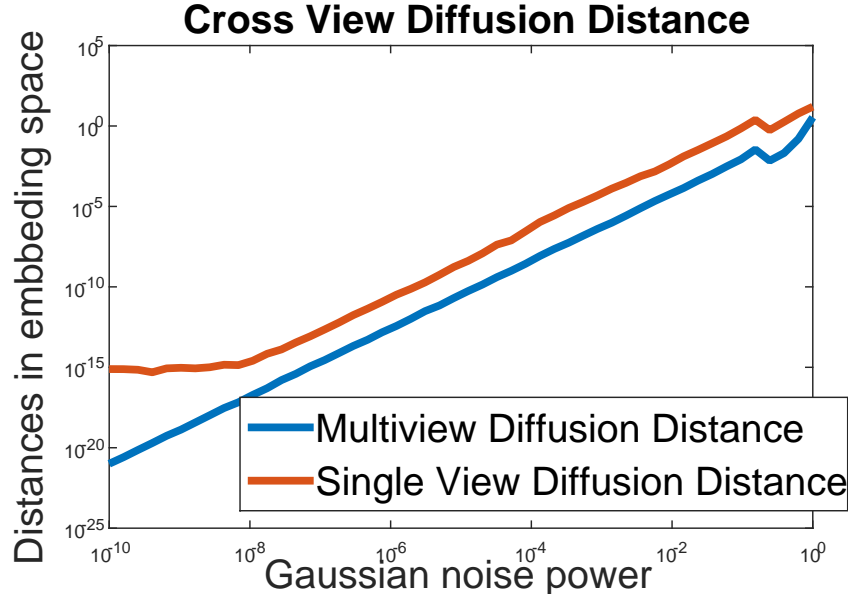


Fig. 7: Comparison between two cross view diffusion based distances. Simulated on two Swiss rolls with additive Gaussian noise. The results are the median of 100 simulations.

extracted. The sum of distances between all the data points in the embedding spaces is denoted as a single view diffusion distance (SVDD). The distance is computed using the following measure

$$\mathcal{D}_t^{(SV)^2}(X, Y) = \sum_{i=1}^M \|\Psi_t(\mathbf{x}_i) - \Psi_t(\mathbf{y}_i)\|^2, \quad (41)$$

where $\Psi_t(\mathbf{x}_i), \Psi_t(\mathbf{y}_i), i = 1, \dots, M$ are the single view diffusion mappings. Then, the proposed framework is applied to extract the coupled embedding. A Cross View Diffusion Distance (CVDD) is computed using Eq. (20). This experiment was executed 100 times for various values of the Gaussian noise variance σ_N^2 .

In about 10% of the single view simulations the embeddings' axis are flipped. This generates a large SVDD although the embeddings share similar structures. In order to remove these type of errors we use the Median of 100 simulations. The median of the results are presented in Fig. 7.

B. Multi-view clustering

The task of clustering has been in the core of machine learning for many years. The goal is to divide a given data set into subsets based on the inherited structure of the data. We use the multi-view construction to extract low dimensional mappings from multiple sets of high dimensional data points. In the following experiments we demonstrate the advantage of the proposed approach on both artificial and real data sets. For the real data sets applying the multi-view approach requires an eigen decomposition of large matrices. To reduce the runtime of experiments we use an approximate matrix decomposition based on sparse random projections [43].

1) *Two circles clustering*: Spectral properties of data sets are useful for clustering since they reveal information about the unknown number of clusters. The characteristic of the eigenvalues of $\hat{\mathbf{P}}$ (Eq. (6)) can provide insight into the number of clusters within the data set. The study in [44] relates the number of clusters to the multiplicity of the eigenvalue 1. A different approach in [45] provides an analysis about the relation between the eigenvalue drop to the number of clusters. In this section, we evaluate how our proposed method captures the clusters' structure when two views are available.

We generate two circles that represent the original clusters using the function

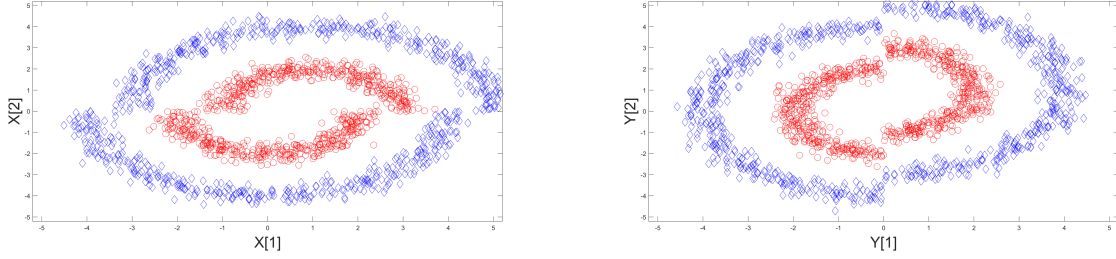


Fig. 8: Left: first view \mathbf{X} . Right: second view \mathbf{Y} . The ground truth clusters are represented by the marker's shape and color.

$$\mathbf{Z} = \begin{bmatrix} z_i[1] \\ z_i[2] \end{bmatrix} = \begin{bmatrix} r \cdot \cos(\theta_i) \\ r \cdot \sin(\theta_i) \end{bmatrix}, \quad (42)$$

where 1600 points $\theta_i, 1 \leq i \leq 1600$, are spread linearly within the line $[0, 4\pi]$. The clusters are created by changing the radius as follows:

$r = 2, 1 \leq i \leq 800$ (first cluster), $r = 4, 801 \leq i \leq 1600$ (second cluster). The views \mathbf{X} (Eq. (43)) and \mathbf{Y} (Eq. (44)) are generated by the application of the following non-linear functions that produce the distorted views

$$x_i[1] = \begin{cases} z_1[i] + 1 + n_i[2] | z_i[2] \geq 0 \\ z_1[i] + n_i[3] | z_i[2] < 0 \end{cases}, x_i[2] = z_i[2] + n_i[1] \quad (43)$$

and

$$y_i[1] = z_i[1] + n_i[4], y_i[2] = \begin{cases} z_i[2] + 1 + n_i[6] | z_i[1] \geq 0 \\ z_i[2] + n_i[6] | z_i[1] < 0 \end{cases}, \quad (44)$$

where $n_i[l], 1 \leq l \leq 6$, are i.i.d random variables drawn from a Gaussian distribution with $\mu = 0$ and $\sigma_n^2 \in [0.03, 0.6]$. This data is referred as the Coupled Circles dataset.

In Fig. 8, the views \mathbf{X} and \mathbf{Y} , which were generated by Eqs. (43) and (44), are presented. Color and shape indicate the ground truth clusters. Initially, DM is applied to each view and clustering is performed using K-means ($K = 2$) within the first diffusion coordinate. The kernel bandwidths σ_x and σ_y for all methods are set using the min-max method described in Eq. (16). We use $t = 1$ since it is optimal for clustering tasks. For the kernel product method we use $\sigma_o = \sqrt{\sigma_x^2 + \sigma_y^2}$. We further extract a 1-dimensional representation using the proposed multi-view framework (Eq. (13)), the Kernel Sum DM (Eq. (9)), Kernel Product DM (Eq. (7)), De Sa's approach (Eq. (37)) and Kernel CCA (Eq. (35)) described in Section V. The regularization parameter is $\gamma = 0.01$ for KCCA and we use 100 components for the Incomplete Cholesky Decomposition [40], [41]. Clustering is performed in the representation space by the application of K-means where $K = 2$. To evaluate the performance of our proposed map 100 simulations with various values of the Gaussian's noise variance (all with zero mean) were performed. The average clustering success rate is presented in Fig. 9. It is evident that the multi-view based approach outperforms the DM-based single view and the Kernel Product approaches.

The performance of kernel methods is highly dependent on setting an appropriate kernel bandwidth σ_x, σ_y , in Algorithm 1 we have presented method for setting such parameters. To evaluate the influence of such parameters on the clustering quality we set $\sigma_n = 0.16$ and extract the multi-view, Kernel Sum and Kernel Product diffusion mapping for various values of σ_x, σ_y . The average clustering performance using Kmeans $K = 2$ are presented in Fig. 10.

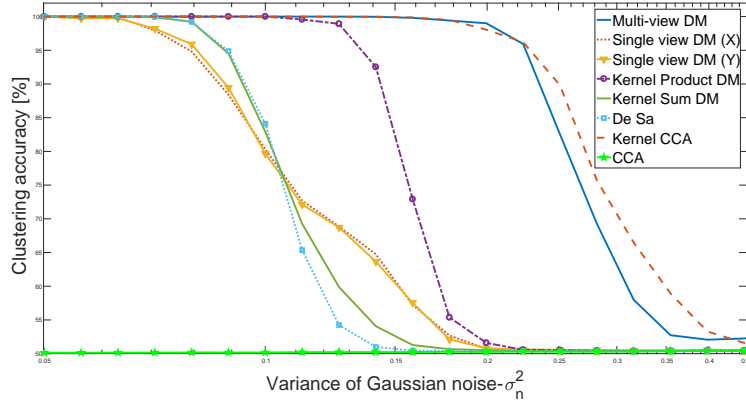


Fig. 9: Clustering results from averaging 200 trials vs. the variance of the Gaussian noise. The simulation performed on the Coupled Circles data (Eqs. (42), (43) and (44)).

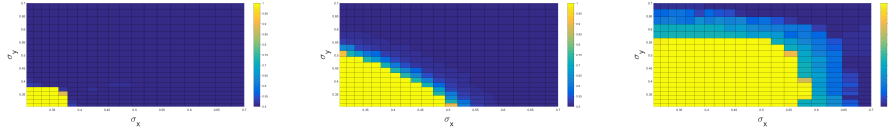


Fig. 10: Clustering results from averaging 20 trails using various values of σ_x, σ_y based on different mappings. The standard deviation of the noise is $\sigma_n = 0.16$. Left- Kernel Sum DM, middle- Kernel Product DM, right- Multi View DM.

2) *Handwritten digits* : For the following clustering experiment, we use the Multiple Features database [46] from the UCI repository. The data set consists of 2000 handwritten digits from 0 to 9 that are equally spread. The features extracted from these images are the profile correlations (FAC), Karhunen-love coefficients (KAR), Zerkine moment (ZER), morphological (MOR), pixel averages in 2×3 windows and the Fourier coefficients (Fou) as our feature spaces $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, \mathbf{X}^4, \mathbf{X}^5, \mathbf{X}^6$ respectively. We apply dimensionality reduction using a single view DM, Kernel Product DM, Kernel Sum DM and the proposed Multi-view. We apply K-means to the reduced mapping using 6 to 20 coordinates. The clustering performance is measured using the Normalized Mutual Information [47] (NMI). Figure 11 presents the average clustering results using K-Means (K=1).

3) *Isolet data set* : The data set was constructed by recording 150 people pronouncing each letter twice for all 26 letters. The feature vector available is a concatenation of the following features: spectral coefficients, contour, sonorant, pre-sonorant and post-sonorant. The authors do not provide the feature's separation, therefore, the dimension of the feature vector is 617. We use a subset of the data with 1599 instances, thus the features space is $\mathbf{X} \in \mathbb{R}^{1599 \times 617}$. To apply the multi-view approach we compute 3 different kernels and fuse them together. The first kernel \mathbf{K}^1 is the standard Gaussian kernel defined in Eq. (4). \mathbf{K}^2 is a Laplacian kernel defined by

$$K_{i,j}^2 \triangleq \exp\left\{\frac{-|\mathbf{x}_i - \mathbf{x}_j|}{\sigma_2}\right\}. \quad (45)$$

The third kernel \mathbf{K}^3 is an exponent with a correlation distance as the affinity measure, given by

$$K_{i,j}^3 \triangleq \exp\left\{\frac{T_{i,j} - 1}{2\sigma^2}\right\}, i, j = 1, \dots, M, \quad (46)$$

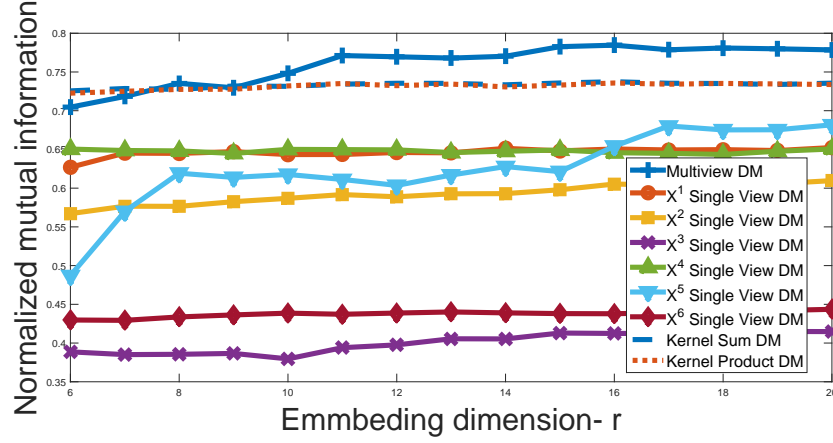


Fig. 11: Average clustering accuracy running 100 simulations on the Handwritten data set. Accuracy is measured using the Normalized Mutual Information (NMI).

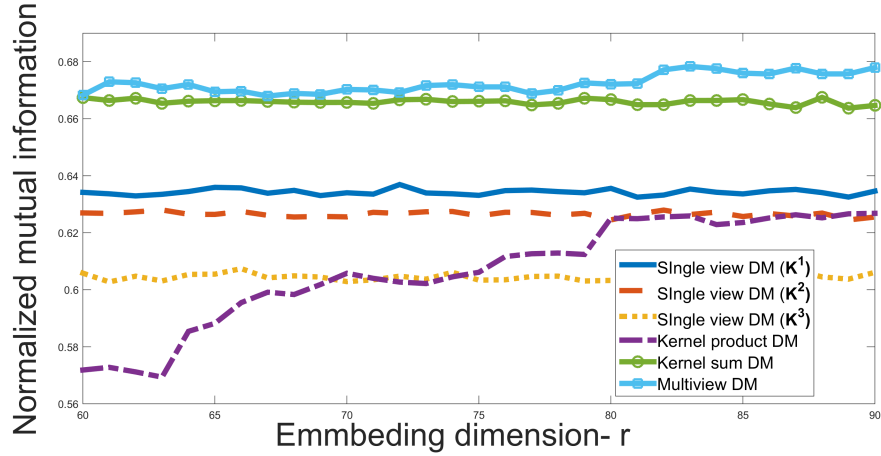


Fig. 12: Clustering accuracy measured with Normalized Mutual Information (NMI) on the Isolet data set by using 3 different kernel matrices. Clustering was performed in the r dimensional embedding space.

where $T_{i,j}$ is the correlation coefficient between the i -th and j -th feature vectors, computed by

$$T_{i,j} \triangleq \frac{\tilde{\mathbf{x}}_i^T \cdot \tilde{\mathbf{x}}_j}{\sqrt{(\tilde{\mathbf{x}}_i^T \cdot \tilde{\mathbf{x}}_i)(\tilde{\mathbf{x}}_j^T \cdot \tilde{\mathbf{x}}_j)}}, i, j = 1, \dots, M. \quad (47)$$

The average subtracted features are $\tilde{\mathbf{x}}_i \triangleq \mathbf{x}_i - \eta_i \cdot \mathbf{1}$, where η_i is the average of the features for instance i . We fuse the kernels using multi-view, kernel product and kernel sum approach, we then apply K-Means to the extracted space. The average NMI for 26 classes is presented in Fig. 12.

C. Manifold learning

The power of manifold learning appears when the data is sampled in a high dimensional space but actually lies on a low dimensional surface. Extracting the underlying surface provides insight into the physical process creating the data. Examples in vision [48], audio [49], medical [50] and more. In this section we demonstrate the proposed approach on an artificial manifold and on a toy example of video sequence.

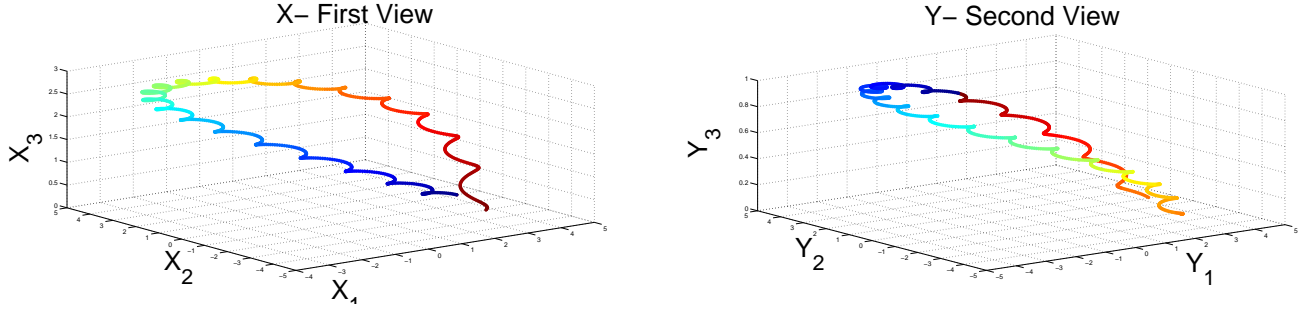


Fig. 13: Top: first Helix \mathbf{X} (Eq. (48)). Bottom: second Helix \mathbf{Y} (Eq. (49)). Both manifolds have some circular structure governed by the angle parameter $a[i]$ and $b[i]$, $i = 1, 2, 3, \dots, 1000$ colored by the points index i .

1) *Artificial manifold learning*: In the general DM approach, there is an assumption that the sampled space describes a low dimensional manifold. However, this assumption can be incorrect since the sampled space can describe the existence of redundancy in the manifold, or more generally, the sampled space can describe two or more manifolds generated by a common physical process. In this section, we examine the extracted embedding computed using our method and compare it to the Kernel Product approach (section III-B).

Helix A

Two coupled manifolds with a common underlying open circular structure are generated. The helix shaped manifolds were generated by the application of a 3 dimensional function to 1000 data points that spread linearly within the lines $a_i \rightarrow [0, 2\pi]$ and $b_i = a_i + 0.5\pi \bmod 2\pi$, $i = 1, 2, 3, \dots, 1000$. The functions in Eqs. (48) and (49) are used to generate the datasets for View-I and View-II denoted as \mathbf{X} and \mathbf{Y} , respectively:

$$\text{View I: } \mathbf{X} = \begin{bmatrix} x_i[1] \\ x_i[2] \\ x_i[3] \end{bmatrix} = \begin{bmatrix} 4 \cos(0.9a_i) + 0.3 \cos(20a_i) \\ 4 \sin(0.9a_i) + 0.3 \sin(20a_i) \\ 0.1(6.3a_i^2 - a_i^3) \end{bmatrix}, i = 1, 2, 3, \dots, 1000, \quad (48)$$

$$\text{View II: } \mathbf{Y} = \begin{bmatrix} y_i[1] \\ y_i[2] \\ y_i[3] \end{bmatrix} = \begin{bmatrix} 4 \cos(0.9b_i) + 0.3 \cos(20b_i) \\ 4 \sin(0.9b_i) + 0.3 \sin(20b_i) \\ 0.1(6.3b_i^2 - b_i^3) \end{bmatrix}, i = 1, 2, 3, \dots, 1000. \quad (49)$$

The 3-dimensional Helix shaped manifolds \mathbf{X} and \mathbf{Y} are presented in Fig. 13.

The Kernel Product mapping (Eq. (7)) separates the manifold to a bow and a point as shown in Fig. 15. This structure neither represents any of the original structures nor reveals the underlying parameters a_i, b_i . On the other hand, our embedding (Eq. (13)) captures the two structures one for each view. As shown in Fig. 14, one structure represents the angle of a_i while the other represents the angle of b_i . The Euclidean distance in the new spaces preserves the mutual relations between data points based on the geometrical relation in both views. Moreover, both manifolds are in the same coordinate system and this is a strong advantage as it enables us to compare between the manifolds in the lower dimensional space. The Euclidean distance in the new spaces preserves the mutual relations between data points that are based on the geometrical structure of both views.

Helix B

The previous experiment was repeated with the functions in Eqs. (50) and (51) to generate datasets for View-I and View-II denoted by \mathbf{X} and \mathbf{Y} , respectively.

$$\text{View I: } \mathbf{X} = \begin{bmatrix} x_i[1] \\ x_i[2] \\ x_i[3] \end{bmatrix} = \begin{bmatrix} 4 \cos(5a_i) \\ 4 \sin(5a_i) \\ 4a_i \end{bmatrix}, \quad (50)$$

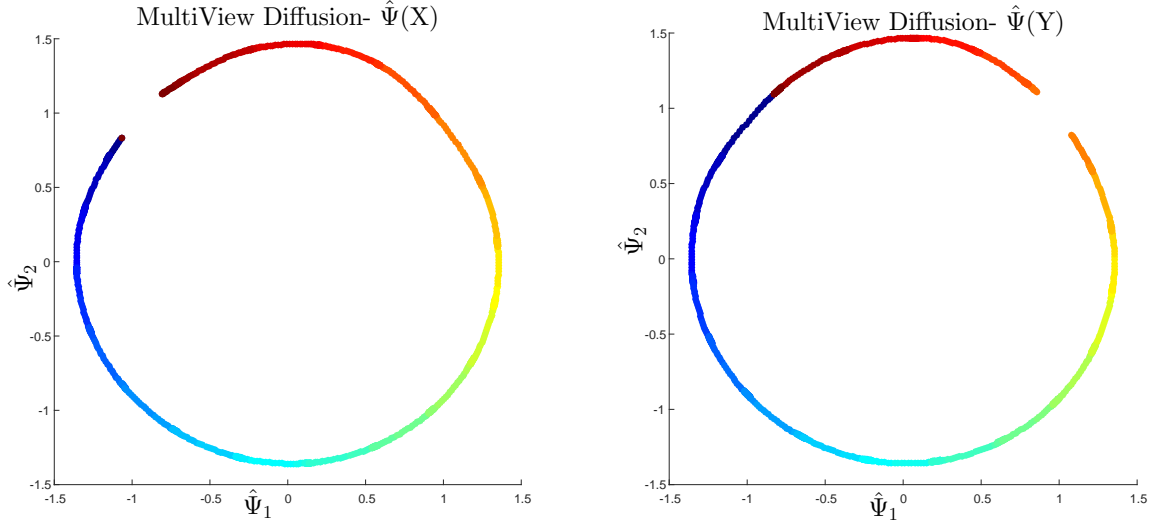


Fig. 14: Top: Multi-View based embedding of the first view $\hat{\Psi}(\mathbf{X})$. Bottom: Multi-View based embedding of the second view $\hat{\Psi}(\mathbf{Y})$. They were computed by using Eq. (13)), respectively.

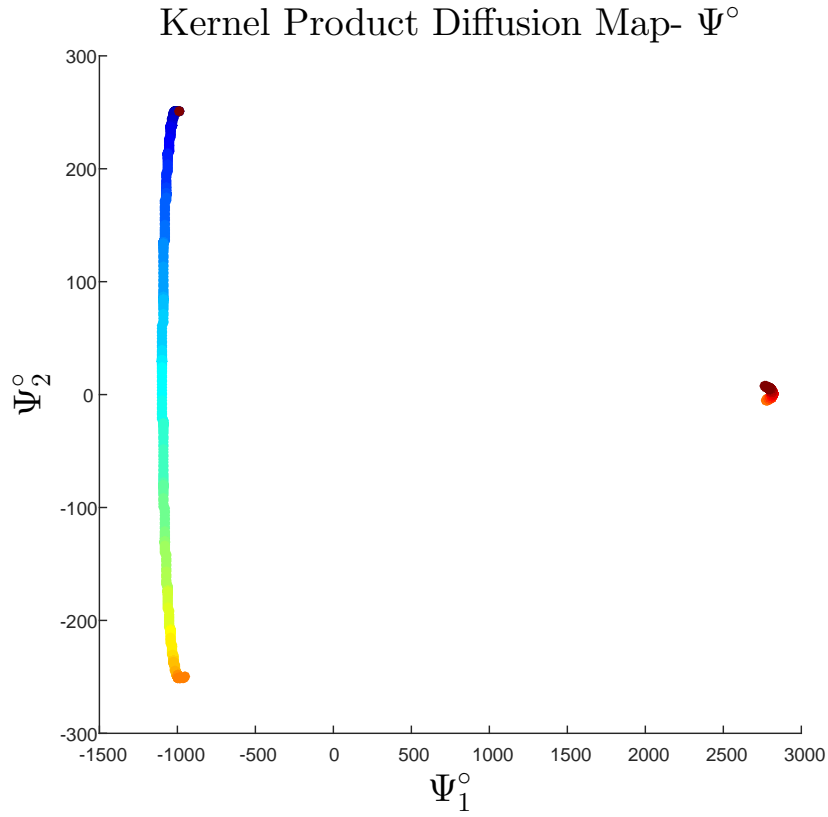


Fig. 15: 2-dimensional DM-based mapping of the Helix computed using the concatenated vector from both views that correspond to the kernel \mathbf{P}° (Eq.(7)).

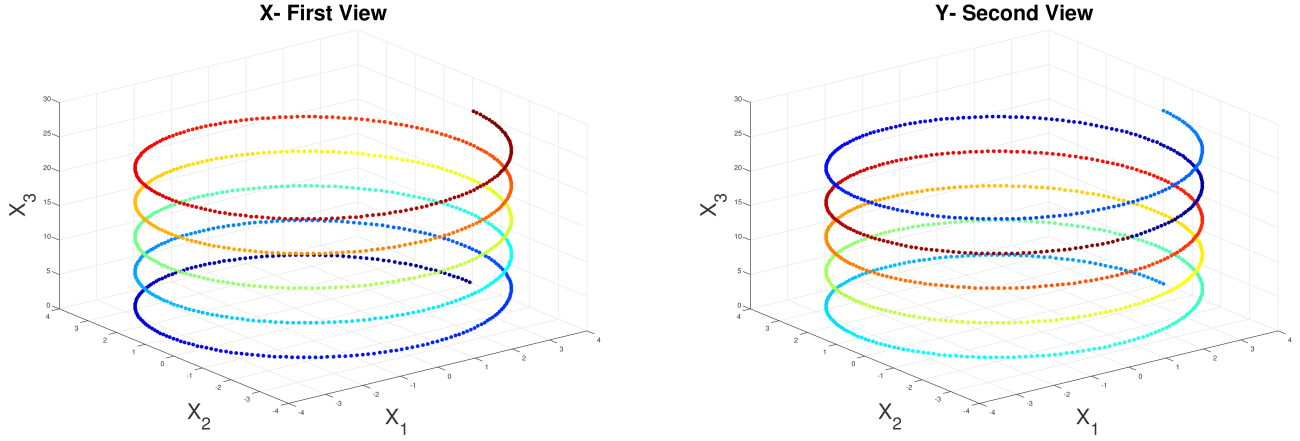


Fig. 16: Top: first Helix \mathbf{X} (Eq. (50)). Bottom: second Helix \mathbf{Y} (Eq. (51)). Both manifolds have some circular structure governed by the angle parameter $a[i]$ and b_i , $i = 1, 2, 3, \dots, 1000$, as colored by the point's index i .

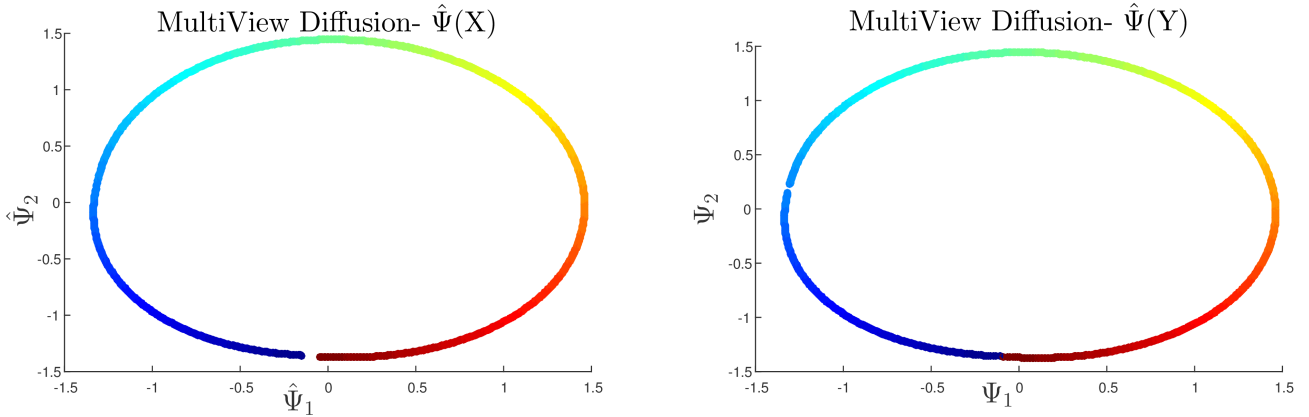


Fig. 17: The coupled mappings computed using our proposed parametrization in Eq. 13

$$\text{View II: } \mathbf{Y} = \begin{bmatrix} y_i[1] \\ y_i[2] \\ y_i[3] \end{bmatrix} = \begin{bmatrix} 4 \cos(5b_i) \\ 4 \sin(5b_i) \\ 4b_i \end{bmatrix}. \quad (51)$$

Again, 1000 points were generated using $a_i \rightarrow [0, 2\pi]$, $b_i = a_i + 0.5\pi \mod 2\pi$, $i = 1, 2, 3, \dots, 1000$. The generated manifolds are presented in Fig. 16.

As can be viewed in Fig. 17, the proposed embeddings (Eq. (13)) has successfully captured the governing parameters a_i and b_i . The Kernel Product based embedding (Eq. (7)) is presented in Fig. 18. The Kernel Product based embedding again separated the data points into two unconnected structures that do not represent well the parameters.

2) *MultiView video sequence*: Various examples such as images, audio, MRI [28], [8] and [51] have demonstrated the power of DM for extracting from real datasets the underlying changing physical parameters. In this experiment, the multi-view approach is tested on a real data. Two web cameras and a toy train with a pre-designed path are used. The train's path has an "eight" shape structure. Extracting the underlying manifold from the set of images enables us to organize the images according to the location along the train's path and thus reveals the true underlying parameters of the processes.

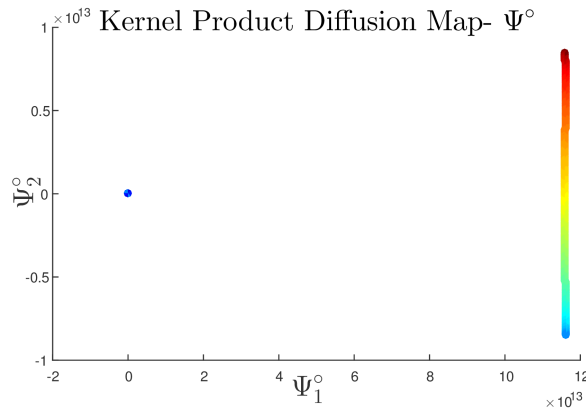


Fig. 18: A 2-dimensional mapping, extracted based on P^o (Eq. (7)).

The setting of the experiment is as follows: each camera records a set of images from a different angle. A sample frame from each view is presented in Fig. 19. The video is sampled at 30 frames per second with a resolution of 640×480 pixels per frame. $M = 220$ images were collected from each view (camera). Then, the R,G,B values were averaged and downsampled to 160×120 pixels resolution. The matrices were reshaped into column vectors. The resulted set of vectors are denoted by \mathbf{X} and \mathbf{Y} where $\mathbf{x}_i, \mathbf{y}_i \in R^{19200}$, $1 \leq i \leq 220$. The sequential order of the images is not important for the algorithm. In a normal setting, one view is sufficient to extract the parameters that govern the movement of the train and thus extract the natural order of the images. However, we use two types of interferences to create a scenario in which each view by itself is insufficient for the extraction of the underlying parameters. The first interference is a gap in the recording of each camera. We remove 20 consecutive frames from each view at different time locations. By doing it, the bijective correspondence of some of the images in the sequence is broken. However, even an approximated correspondence is sufficient for our proposed manifold extraction. A standard 2-dimensional DM base mapping of each view was extracted. The results are bow shaped manifolds as presented in Fig. 20. Applying DM separately to each view extracts the correct order of the data points (images) along the path. However, the “missing” data points broke the circular structure of the expected manifold and resulted in a bow shaped embedding. We use the multi-view based methodology to overcome this interference by application of the multi-view framework to extract two coupled mapping (Eq. (13)). The results are presented in Fig. 21. The proposed approach overcomes the interferences by smoothing the gap inherited in each view through the use of connectivities from the “undistracted” view. Finally, we concatenate the vectors from both views and compute the Kernel Product embedding. The results are presented in Fig. 22. Again, the structure of the manifold is distorted and incomplete due to the missing images.

This experiment was repeated while replacing 10 frames from each view with a Gaussian noise that has the parameters $\mu = 0$ and $\sigma^2 = 10$ that are average and variance, respectively. A single view DM-based mapping was computed. The Kernel Product-based DM and the multi-view based DM mappings were computed as well. As presented in Fig. 23, the Gaussian noise distorted the manifolds extracted in each view. The multi-view approach extracted two circular structures presented in Fig. 24. Again, the data points are ordered according to the position along the path. This time, the circular structure is unfolded and the gaps are visible in both embeddings. Applying the Kernel Product approach (Eq. 7) has yielded a distorted manifold as presented in Fig. 25.

D. Classification in the embedding space

Besides improving classification results, dimensionality reduction can reduce the execution time. Studies such as [52], [53] and [54] focus on the role of dimensionality reduction for classification. Applications



Fig. 19: Top: a sample image from the first camera (\mathbf{X}). Bottom: a sample image from the second camera (\mathbf{Y}).

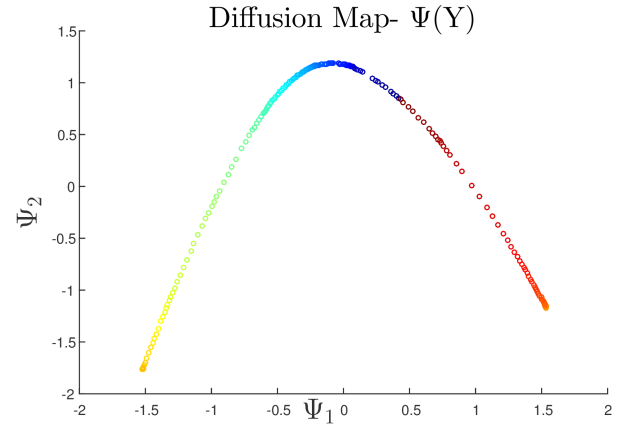
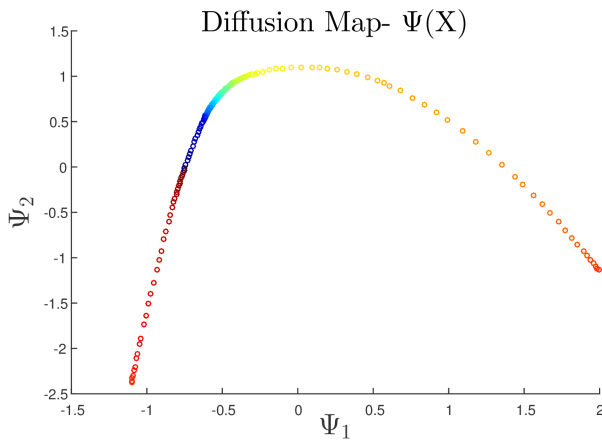


Fig. 20: Top: DM-based single view mapping $\Psi(\mathbf{X})$. Bottom: DM-based single view mapping $\Psi(\mathbf{Y})$. The removed images caused a bow shaped structure.

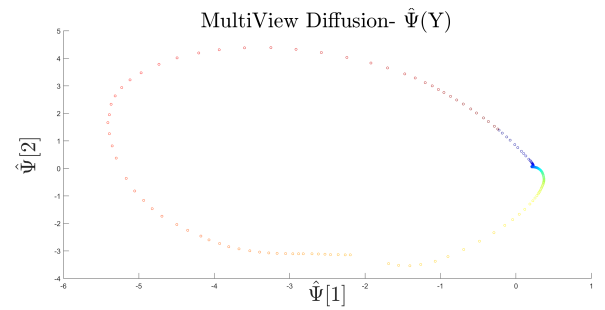
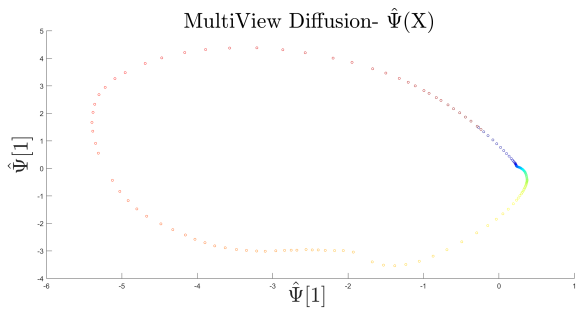


Fig. 21: Top: Mapping $\hat{\Psi}(\mathbf{X})$. Bottom: Mapping $\hat{\Psi}(\mathbf{Y})$ as extracted by the multi-view based framework. Two small gaps, which correspond to the removed images, are visible.

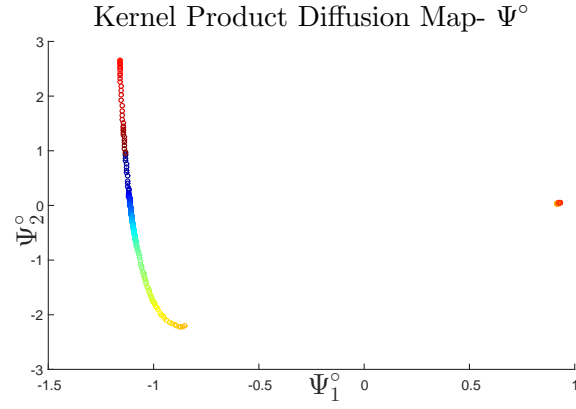


Fig. 22: A standard diffusion mapping (Kernel Product-based) that was computed by using the concatenated vector from both views that correspond to kernel \mathbf{K}° .

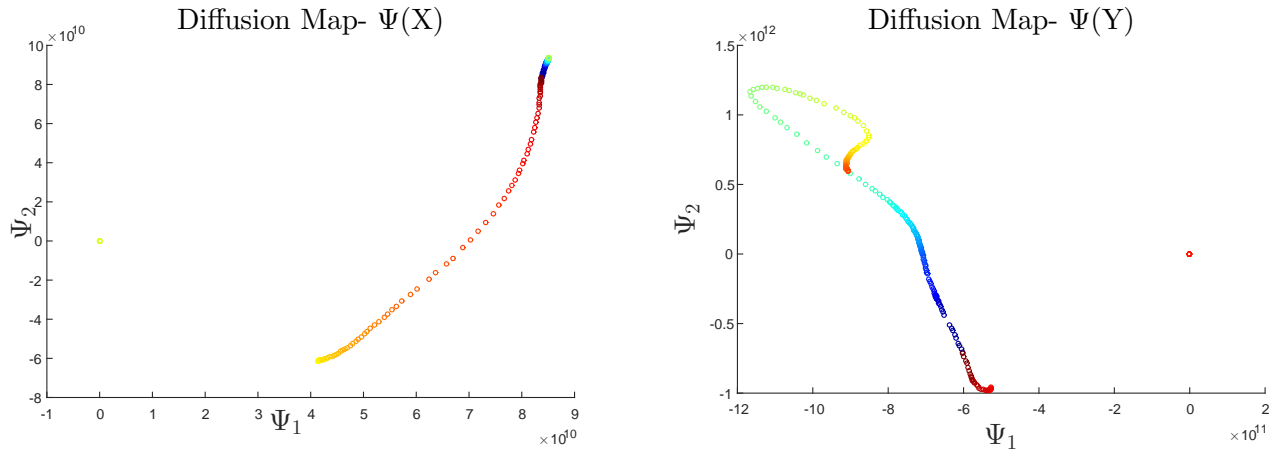


Fig. 23: Top: DM-based single view mapping $\Psi(\mathbf{X})$. Bottom: DM-based single view mapping $\Psi(\mathbf{Y})$. The Gaussian noise deformed the circular structure

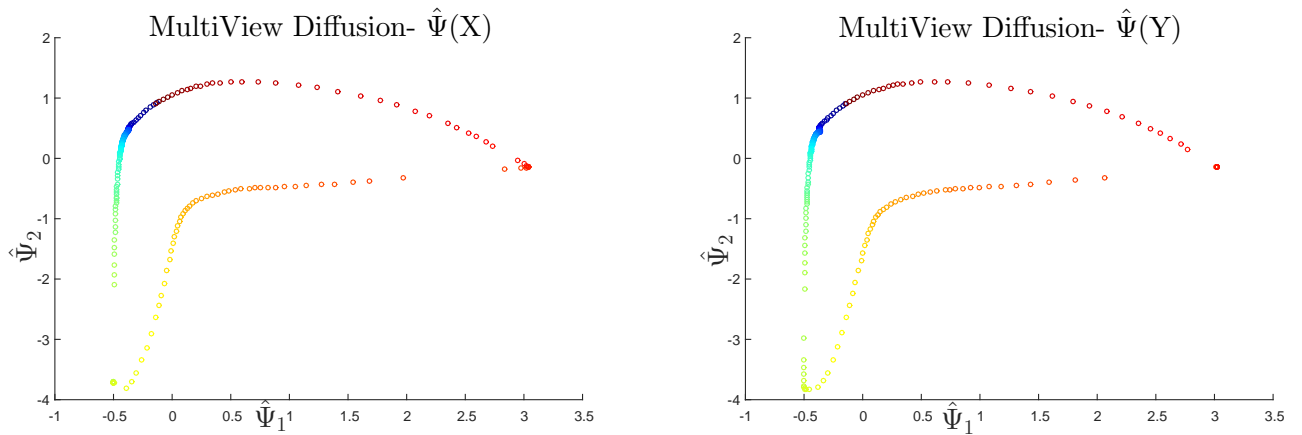


Fig. 24: Top: Mapping $\hat{\Psi}(\mathbf{X})$. Bottom: Mapping $\hat{\Psi}(\mathbf{Y})$ as extracted by the multi-view framework. Two gaps are visible that correspond to Gaussian noise.

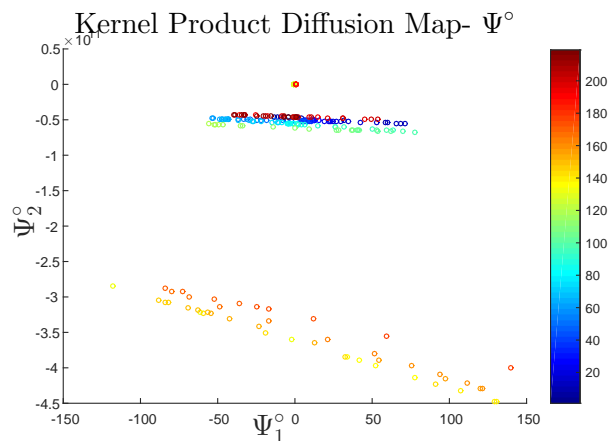


Fig. 25: Computation of a standard diffusion mapping (Kernel Product) by using the concatenation vector from both views (corresponding to kernel P° Eq.(7)).

TABLE I: Classification accuracy using 1-fold cross validation. r is the number of coordinates used in the embedding space.

Method	Accuracy [%] ($r = 3$)	Accuracy [%] ($r = 4$)
Single View DM (\mathbf{X}^1)	89.9	93
Single View DM (\mathbf{X}^2)	88.6	92.4
Single View DM (\mathbf{X}^3)	89.3	91.1
Single View DM (\mathbf{X}^4)	89.3	89.2
Single View DM (\mathbf{X}^5)	89.3	90.5
Single View DM (\mathbf{X}^6)	88.6	91.1
Kernel Sum DM	93.7	94.9
Kernel Product DM	94.3	93
Multi-view DM	97.5	98.1

have been studied in diverse fields [55] [8] [56].

1) *Classification of seismic events*: Various studies have used machine learning for seismic events classification. Some of the methods used are: artificial neural networks [57], [58], [59], self-organizing maps [60], [61], hidden Markov models [62], [63] and support vector machines [64]. We use a data set collected from a seismic catalog by the Geophysical Institute of Israel. The data set includes 46 earthquakes, 62 explosions and 62 noise waveforms. Each waveform was sampled at a frequency of 40 Hz. The length of each waveform is one minute, thus each consists of 2400 samples. Data was collected from two stations. Each station uses a three component seismometer concluding a total number of 6 views.

The features extracted from the waveform are termed Sonograms [65]. First, a Spectrogram is computed by using the short time Fourier transform (STFT). The frequency scale is rearranged to be equally tempered on a logarithmic scale, such that the final spectrogram contains 11 frequency bands. The bins are normalized such that the sum of energy in every frequency band is equal to 1. The resulting set of sonograms denoted $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, \mathbf{X}^4, \mathbf{X}^5, \mathbf{X}^6$. These are the input views for our framework.

We apply the proposed framework, as well as the single view DM, kernel product DM and kernel sum DM. Classification is performed by using K-NN ($K=1$), based on 3 or 4 coordinates from the reduced mapping. The results are presented in table I.

To evaluate the multi-view classification performance on subsets of the $L = 6$ views we use subsets of only two view. We evaluate the classification results based on all the pairs of view $\mathbf{X}^l, \mathbf{X}^m, l, m = 1, \dots, 6, l \neq m$. The accuracy of classification based on the multi-view representation $\hat{\Psi}(\mathbf{X}^l), l = 1, \dots, 6$ given that $\mathbf{X}^m, m = 1, \dots, 6, m \neq l$ is presented in Fig. 26.

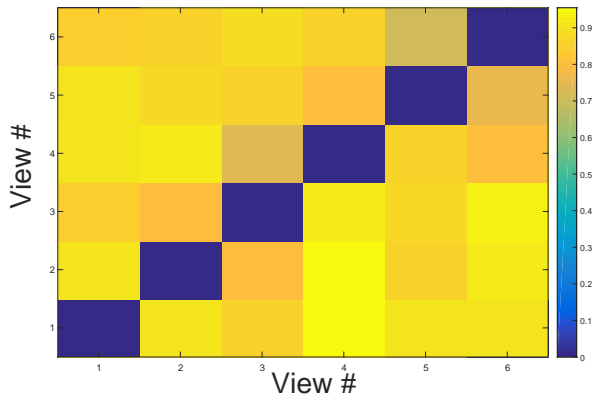


Fig. 26: Classification accuracy using K-nn ($K=1$) for all pairs of views $\mathbf{X}^l, \mathbf{X}^m, l \neq m$. The y-axis is the number of the first view used, while the x-axis is the number of the second view. Classification is performed in the multi-view low dimensional embedding ($r = 4$). The diagonal terms are presented as zero since we did not simulate for $l = m$.

VII. DISCUSSION

In this paper, we presented a framework for dimensionality reduction that is multi-view based. The method enables us to extract simultaneous embeddings from coupled embeddings. We enforce a cross domain probabilistic model at a single time step. The transition probabilities depend on the connectivities in both views. We derived various theoretical aspects of the proposed method and demonstrated their applicabilities to both artificial and real data. The experimental results demonstrate the strength of the proposed framework in cases where data is missing in each view or each of the manifolds is deformed by an unknown function. The framework is applicable to various real life machine learning tasks that consist of multiple views or multiple modalities.

REFERENCES

- [1] I. Jolliffe, *Principal component analysis*, 2005, vol. 21.
- [2] J. B. Kruskal and W. M., "Multidimensional scaling," *Sage Publications. Beverly Hills*, 1977.
- [3] S. T. Roweis and L. K. Sau, "Nonlinear dimensionality reduction by local linear embedding," *Science*, vol. 290.5500, pp. 2323–2326, 2000.
- [4] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *NIPS*, vol. 14, no. 14, 2001, pp. 585–591.
- [5] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.
- [6] W. Luo, "Face recognition based on laplacian eigenmaps," 2011, pp. 416 – 419.
- [7] A. Singer and R. R. Coifman, "Non linear independent component analysis with diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 25(2), pp. 226–239, 2008.
- [8] O. Lindenbaum, A. Yeredor, and I. Cohen, "Musical key extraction using diffusion maps," *Signal Processing*, 2015.
- [9] W. T. Freeman and J. B. Tenenbaum, "Learning bilinear models for two-factor problems in vision," in *Computer Vision and Pattern Recognition, Proceedings., IEEE Computer Society Conference on.* IEEE, 1997, pp. 554–560.
- [10] I. S. Helland, "Partial least squares regression and statistical models," *Scandinavian Journal of Statistics*, pp. 97–114, 1990.
- [11] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proceedings of the 26th annual international conference on machine learning.* ACM, 2009, pp. 129–136.
- [12] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Advances in Neural Information Processing Systems*, 2011, pp. 1413–1421.
- [13] D. Zhou and C. Burges, "Spectral clustering and transductive learning with multiple views," *Proceedings of the 24th international conference on Machine learning*, pp. 1159–1166, 2007.
- [14] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu, "Unsupervised metric fusion by cross diffusion," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 2997–3004.
- [15] A. Kumar and H. Daumé, "A co-training approach for multi-view spectral clustering," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 393–400.
- [16] B. Boots and G. Gordon, "Two-manifold problems with applications to nonlinear system identification," 2012.

- [17] H. Hotelling, "Relations between two sets of variates," *Biometrika*, pp. 321–377, 1936.
- [18] V. R. de Sa, "Spectral clustering with two views," in *ICML workshop on learning with multiple views*, 2005.
- [19] W. Wang and M. A. Carreira-Perpinán, "The role of dimensionality reduction in classification," in *AAAI*, 2014, pp. 2128–2134.
- [20] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Nips*, vol. 2, 2003, p. 5.
- [21] R. Navaratnam, A. W. Fitzgibbon, and R. Cipolla, "The joint manifold model for semi-supervised multi-valued regression," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.
- [22] A. P. Shon, K. Grochow, A. Hertzmann, and R. P. Rao, "Learning shared latent structure for image synthesis and robotic imitation," *Advances in neural information processing systems*, vol. 18, p. 1233, 2006.
- [23] C. H. Ek, P. H. Torr, and N. D. Lawrence, "Gaussian process latent variable models for human pose estimation," in *International workshop on machine learning for multimodal interaction*. Springer, 2007, pp. 132–143.
- [24] L. Sigal, R. Memisevic, and D. J. Fleet, "Shared kernel information embedding for discriminative inference," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2852–2859.
- [25] A. Klami, S. Virtanen, and S. Kaski, "Bayesian canonical correlation analysis," *Journal of Machine Learning Research*, vol. 14, no. Apr, pp. 965–1003, 2013.
- [26] A. Klami and S. Kaski, "Probabilistic approach to detecting dependencies between data sets," *Neurocomputing*, vol. 72, no. 1, pp. 39–46, 2008.
- [27] M. Salzmann, C. H. Ek, R. Urtasun, and T. Darrell, "Factorized orthogonal latent spaces," in *AISTATS*, 2010, pp. 701–708.
- [28] S. Lafon, Y. Keller, and R. Coifman, "Data fusion and multicue data matching by diffusion maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28 no. 11, p. 17841797, 2006.
- [29] M. Hirn and R. Coifman, "Diffusion maps for changing data," *Applied and computational harmonic analysis*, 2013.
- [30] S. Lafon, "Diffusion maps and geometric harmonics," *Ph.D dissertation Yale*, 2004.
- [31] Y. Keller, R. R. Coifman, S. Lafon, and S. W. Zucker, "Audio-visual group recognition using diffusion maps," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 403–413, 2010.
- [32] A. Singer, R. Erban, I. G. Kevrekidis, and R. R. Coifman, "Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps," *Proceedings of the National Academy of Sciences*, vol. 106, no. 38, pp. 16090–16095, 2009.
- [33] Y. Shkolinsky. Lecture notes on spectral methods in data analysis, <https://sites.google.com/site/yoelshkolnisky/teaching>.
- [34] T. Ando, "Majorization relations for hadamard products," *Linear Algebra and its Applications*, pp. 57–64, 1995.
- [35] G. Visick, "A weak majorization involving the matrices a , b and ab ," *Linear Algebra and its Applications*, vol. 224/224, pp. 731–744, 1995.
- [36] M. Belkin and P. Niyogi, "Convergence of laplacian eigenmaps," in *NIPS*, 2006, pp. 129–136.
- [37] M. Hein, J.-Y. Audibert, and U. Von Luxburg, "From graphs to manifolds—weak and strong pointwise consistency of graph laplacians," in *International Conference on Computational Learning Theory*. Springer, 2005, pp. 470–485.
- [38] A. Singer, "From graph to manifold laplacian: The convergence rate," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 128–134, 2006.
- [39] R. R. Lederman and R. Talmon, "Common manifold learning using alternating-diffusion," submitted, Tech. Report YALEU/DCS/TR1497, Tech. Rep., 2014.
- [40] P. L. Lai and C. Fyfe, "Kernel and nonlinear canonical correlation analysis," *International Journal of Neural Systems*, vol. 10, no. 05, pp. 365–377, 2000.
- [41] S. Akaho, "A kernel method for canonical correlation analysis," *arXiv preprint cs/0609071*, 2006.
- [42] D. S. Kershaw, "The incomplete choleskyconjugate gradient method for the iterative solution of systems of linear equations," *Journal of Computational Physics*, vol. 26, no. 1, pp. 43–65, 1978.
- [43] Y. Aizenbud and A. Averbuch, "Matrix decompositions using sub-gaussian random matrices," *arXiv preprint arXiv:1602.03360*, 2016.
- [44] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," *Proceedings of Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, vol. 14, pp. 849–856, 2001.
- [45] M. Polito and P. Perona, "Grouping and dimensionality reduction by locally linear embedding," in *NIPS*, 2001, pp. 1255–1262.
- [46] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [47] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189–201, 2009.
- [48] A. Elgammal and C.-S. Lee, "Inferring 3d body pose from silhouettes using activity manifold learning," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–681.
- [49] R. Talmon, I. Cohen, and S. Gannot, "Relative transfer function identification using convolutive transfer function approximation," *IEEE Transactions on audio, speech, and language processing*, vol. 17, no. 4, pp. 546–555, 2009.
- [50] C. Wachinger, M. Yigitsoy, E.-J. Rijkhorst, and N. Navab, "Manifold learning for image-based breathing gating in ultrasound and mri," *Medical image analysis*, vol. 16, no. 4, pp. 806–818, 2012.
- [51] G. Piella, "Diffusion maps for multimodal registration," *Sensors*, vol. 14, no. 6, pp. 10562–10577, 2014.
- [52] W. Wang and M. A. Carreira-Perpinán, "The role of dimensionality reduction in classification," in *AAAI*, 2014, pp. 2128–2134.
- [53] T. S. Tian, "Dimensionality reduction for classification with high-dimensional data," Ph.D. dissertation, Citeseer, 2009.
- [54] F. Plastria, S. De Bruyne, and E. Carrizosa, "Dimensionality reduction for classification," in *International Conference on Advanced Data Mining and Applications*. Springer, 2008, pp. 411–418.
- [55] B. Li, C.-H. Zheng, D.-S. Huang, L. Zhang, and K. Han, "Gene expression data classification using locally linear discriminant embedding," *Computers in Biology and Medicine*, vol. 40, no. 10, pp. 802–810, 2010.
- [56] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

- [57] T. Tiira, "Discrimination of nuclear explosions and earthquakes from teleseismic distances with a local network of short period seismic stations using artificial neural networks," *Physics of the earth and planetary interiors*, vol. 97, no. 1, pp. 247–268, 1996.
- [58] E. Del Pezzo, A. Esposito, F. Giudicepietro, M. Marinaro, M. Martini, and S. Scarpetta, "Discrimination of earthquakes and underwater explosions using neural networks," *Bulletin of the Seismological Society of America*, vol. 93, no. 1, pp. 215–223, 2003.
- [59] A. Esposito, F. Giudicepietro, S. Scarpetta, L. DAuria, M. Marinaro, and M. Martini, "Automatic discrimination among landslide, explosion-quake, and microtremor seismic signals at stromboli volcano using neural networks," *Bulletin of the Seismological Society of America*, vol. 96, no. 4A, pp. 1230–1240, 2006.
- [60] B. Sick, M. Guggenmos, and M. Joswig, "Chances and limits of single-station seismic event clustering by unsupervised pattern recognition," *Geophysical Journal International*, vol. 201, pp. 1801–1813, 2015.
- [61] A. K'ohler, M. Ohrnberger, and F. Scherbaum, "Unsupervised pattern recognition in continuous seismic wavefields records using self-organizing maps," *Geophysical Journal International*, vol. 185, pp. 1619–1630, 2010.
- [62] M. Beyreuther, C. Hammer, M. Wassermann, M. Ohrnberger, and M. Megies, "Constructing a hidden markov model based earthquake detector: Application to induced seismicity," *Geophysical Journal International*, vol. 189, pp. 602–610, 2012.
- [63] C. Hammer, M. Ohrnberger, and D. F'ah, "Classifying seismic waveforms from scratch: A case study in the alpine environment," *Geophysical Journal International*, vol. 192, pp. 425–439, 2013.
- [64] J. Kortstr'om, U. Marja, and T. Tiira, "Automatic classification of seismic events within a regional seismograph network," *Comuters and Geosciences*, vol. 87, pp. 22–30, 2016.
- [65] J. H. Kurz, C. U. Grosse, and H.-W. Reinhardt, "Strategies for reliable automatic onset time picking of acoustic emissions and of ultrasound signals in concrete," *Ultrasonics*, vol. 43, no. 7, pp. 538–546, 2005.